Mixture-of-Control: State-Aware Fine-Tuning for Transformer-based Models

Duc-Anh Nguyen[1,2,*†‡] **Tien Ngoc Luu**[1,*] **Tung Pham**[1] **Toan Tran**[1,‡]

[1]Qualcomm AI Research[§] [2]University of Virginia

Abstract

State-based fine-tuning has emerged as a compelling alternative to weight-based adaptation for transformers, updating lightweight controls into states rather than model weights, offering substantial memory savings while retaining parameter efficiency. However, most existing state-based methods typically apply only per-block control updates, which limits inter-block information exchange and restricts representational adaptation. Meanwhile, prior mechanisms that enable cross-block communication often introduce considerable computational overhead, reducing their practicality for efficient fine-tuning. We introduce Mixture-of-Control (MoC), an efficient cross-layer communication that adaptively integrates local and global control signals to enhance representation learning during adaptation. MoC treats block-wise control states as experts in a sparse mixture-of-experts process, enabling efficient communication across transformer blocks. Experiments on 33 datasets and 10 pretrained models across diverse transformer-based benchmarks demonstrate that MoC consistently outperforms existing state-based fine-tuning methods while maintaining comparable efficiency in memory and compute.

1 Introduction

Fine-tuning large pretrained models is a standard approach to adapt foundation models to downstream tasks, but full fine-tuning is expensive: it updates and stores billions of parameters per task. Parameter-efficient fine-tuning (PEFT) reduces this cost by freezing the backbone and learning only a small set of task parameters. Among PEFT methods, LoRA [1] is particularly effective, replacing weight updates with a low-rank decomposition that largely preserves performance while sharply reducing trainable parameters.

Many LoRA variants–including VeRA [2], BinaryLoRA [3], MoLE [4], and DoRA [5]–push parameter or compute efficiency via vectorized updates, binary adaptation, expert mixtures, or decomposed updates. However, most remain layer-local, offering limited cross-layer communication and constraining structured representation adaptation. MoLEx [6] partially addresses this with a Sparse Mixture-of-Layer-Experts, but its Top-K selection requires extra forward passes, raising per-layer cost, especially in large models.

Beyond weak cross-layer coordination, many methods optimize parameter count but overlook training-time memory. For example, halving DoRA's rank reduces GPU memory by only 0.256GB on an 8B model [7], showing that fewer trainable parameters need not yield meaningful memory savings. Similarly, existing cross-layer interaction mechanisms such as MoLEx [6] often add substantial compute and memory overhead, limiting scalability.

*Equal contribution.

†This work was done during Duc-Anh Nguyen's internship at Qualcomm.

‡Corresponding authors: nguyenducanh909.bkhn@gmail.com, toatran@qti.qualcomm.com.

§Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

These two issues–restricted cross-layer communication and high memory overhead–motivate a different perspective: fine-tuning via control theory [8]. Rather than modifying system parameters, control systems steer states via feedback. Parallel Control [7] casts LoRA as a control term and activations as states, reducing memory by avoiding intermediate-state storage. Yet it remains local to each layer and lacks global coordination across depth. This motivates our central research question:

"How can state-based fine-tuning be designed to capture deep cross-layer interactions without sacrificing its core advantage: memory efficiency?"

We answer this with **Mixture-of-Control (MoC)**, a lightweight state-based method that couples local and global signals via a recurrent mixture-of-experts control pathway. MoC treats each layer's low-rank control module as an *expert* and uses a shared gate to sparsely route each layer's representation to a Top-K set of experts, producing a global control signal. This global signal is then mixed with the layer-local control (e.g., α vs. $1-\alpha$) and injected into the frozen block, enabling adaptive information propagation across depth. Our MoC supports cross-layer communication via low-rank projections without extra pretrained-block passes, preserving state-memory efficiency with minimal overhead. It is architecture-agnostic (FFN or attention; shared or separate gates) and improves robustness and adaptation across benchmarks while matching PEFT-level memory use.

In summary, our contributions are fourfold:

- We identify key limits of existing PEFT and state-based adaptation: controllers are mostly block-local (weak global coordination), while cross-block mechanisms often add substantial, block-dependent compute that impedes scaling to deep/complex architectures.
- We show that cross-block communication can be realized via a lightweight low-rank *control* pathway that exchanges information across layers without reusing pretrained blocks, offering an efficient alternative to full-block interaction.
- We propose **Mixture-of-Control (MoC)**, an efficient cross-layer communication framework that integrates block-local control with a sparsely routed global mixture of control experts to enhance the representation learning, supported by theory showing expressiveness preservation and improved stability/gradient flow/representation error under mild assumptions.
- We conduct extensive experiments across 33 benchmarks spanning NLU, NLG, and vision tasks on 10 pretrained Transformer backbones, showing that MoC consistently improves adaptation while matching strong PEFT baselines in parameter and memory efficiency.

2 Related Works

Parameter-Efficient Fine-Tuning (PEFT) The rapid growth of Transformer-based models [9, 10] has made full fine-tuning costly, motivating PEFT. Early PEFT methods tune selected layers [11–14], while later approaches add lightweight trainable components. Adapter-based methods insert bottleneck modules [15, 16], and prompt-based methods optimize soft prompts [17–19], though they are often sensitive to initialization [20].

Low-Rank Adaptation (LoRA) [1] reduces the number of trainable parameters by modeling weight updates with low-rank matrices. Subsequent variants further improve efficiency through parameter sharing, quantization, or adaptive rank/parameter selection [2, 21–23], while extensions such as MoLE and DoRA (which separates magnitude and direction) further enhance adaptability and performance [4, 5, 24–26]. However, most methods remain layer-local, thereby limiting cross-layer information exchange during adaptation.

Cross-Layer Communication for Adaptation To overcome the layer-local limitations of LoRA-based methods, Mixture of Layer Experts (MoLEx) [6] treats each pretrained layer as an expert and performs TopK routing across depth to aggregate cross-layer information. Although effective, this strategy requires additional forward passes through selected layers, which incurs substantial computational overhead, particularly in deep or complex architectures.

Control-Theoretic View and State-Based Tuning PEFT can be viewed through control theory [27], treating deep networks as dynamical systems evolving feature states across layers [28]. This view motivates work on continuous-depth models, stability, and optimization [29–32]. Zhang et al. [8]

cast LoRA as a control input over nominal pretrained dynamics, and [7] introduce Parallel Control to reduce memory by skipping intermediate states. We extend this framework with a scalable state-based adaptation method that improves performance while preserving similar memory efficiency.

3 Preliminaries

3.1 Weight-based Fine-Tuning

LoRA [1] fine-tunes $W_0 \in \mathbb{R}^{d\times k}$ by adding a low-rank update $\Delta W = BA$ with $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $r \ll \min\{d, k\}$, yielding $W' = W_0 + BA$ where W_0 is frozen and A, B are trainable. Many variants exploit this structure for efficiency, including VeRA [2], MoLE [4], and DoRA [5], which leverage low-rank structures to reduce computational cost. These mechanisms are categorized as weight-based fine-tuning (Weight-FT), which adapts model weights directly. Its effectiveness is often attributed to the low intrinsic-rank structure of task-specific updates [1, 33]. Mathematically, let x_0 be the input, x_t be the output representation at block t, and a_t be a nonlinear activation (ReLU or GeLU), a pretrained transformer block can be adapted as follows [34]:

$$x_{t+1} = x_t + a_t((W_t + \Delta W_t)x_t)\,. \tag{1}$$

Weight-FT often incurs high memory costs due to storing intermediate states. To mitigate this, the State-Based Framework [7] reformulates fine-tuning from a control-theoretic perspective.

3.2 State-based Fine-Tuning

Forward Pass as the Feedback Control Consider the Ordinary Differential Equation system [35]:

$$\dot{x}(t) = f_t(W(t)x(t)) + G(t)u(t), \tag{2}$$

where $x(t)$ denotes the state, $W(t)$ the state matrix, f_t the intrinsic dynamics, and $G(t)u(t)$ the control term. Using state feedback $u(t) = K(t)x(t)$ with feedback gain $K(t)$, Equation (2) becomes

$$\dot{x}(t) = f_t(W(t)x(t)) + G(t)K(t)x(t). \tag{3}$$

Discretizing Equation (3) yields

$$x_{t+1} = x_t + f_t(W_t x_t) + G_t K_t x_t. \tag{4}$$

This matches a neural-network forward pass when x_t is the t-block state representation.

State-based Fine-Tuning Motivated by this analogy, Zhang et al. [7] introduced Parallel Control, interpreting $G_t K_t$ as a low-rank control term, i.e.,

$$x_{t+1} = x_t + f_t(W_t x_t) + g_t(x_t), \tag{5}$$

where f_t is the pretrained block (e.g., attention/ FFN), $g_t(x_t) = \Delta W_t x_t$ is implemented by a low-rank adapter. This view retains efficient adaptation while reducing intermediate-memory overhead.

3.3 Sparse Mixture of Experts (SMoE)

A SMoE layer contains n experts and a gating network to process different aspects of the input. Given m token embeddings $\mathbf{X} \in \mathbb{R}^{m\times d}$ and gate weights $\mathbf{W}_{\text{gate}} \in \mathbb{R}^{n\times d}$, the routing scores are $\mathbf{S} = \mathbf{X}\mathbf{W}_{\text{gate}}^{\top} \in \mathbb{R}^{m\times n}$. For each token $\mathbf{x}_i \in \mathbb{R}^d$, only the top-$k$ expert indices $\mathcal{T}_i \subseteq \{1, \ldots, n\}$ with $|\mathcal{T}_i| = k$ are activated. Their normalized routing weights are $w_{i,j} = \begin{cases} \dfrac{\exp(s_{i,j})}{\sum_{j'\in\mathcal{T}_i}\exp(s_{i,j'})} & \text{, if } j \in \mathcal{T}_i, \\ 0 & \text{, otherwise.} \end{cases}$

Each expert $e_j : \mathbb{R}^d \to \mathbb{R}^d$ is presented by a feedforward network, then the output is aggregated as

$$\mathbf{y}_i = \sum_{j\in\mathcal{T}_i} w_{i,j} \cdot e_j(\mathbf{x}_i). \tag{6}$$

To mitigate routing collapse [36–38] (where a few experts dominate token assignments), a commonly used auxiliary load-balancing loss is introduced: $\mathcal{L}_{\text{Balance}} = \lambda \sum_{j=1}^{n} \tau_j P_j$, where λ is a hyperparameter, with $P_j = \frac{1}{m}\sum_{i=1}^{m} \text{softmax}(\mathbf{s}_{i,:})_j$ and $\tau_j = \frac{n}{km}\sum_{i=1}^{m} \mathbf{1}(\text{token } i \text{ selects expert } j)$.

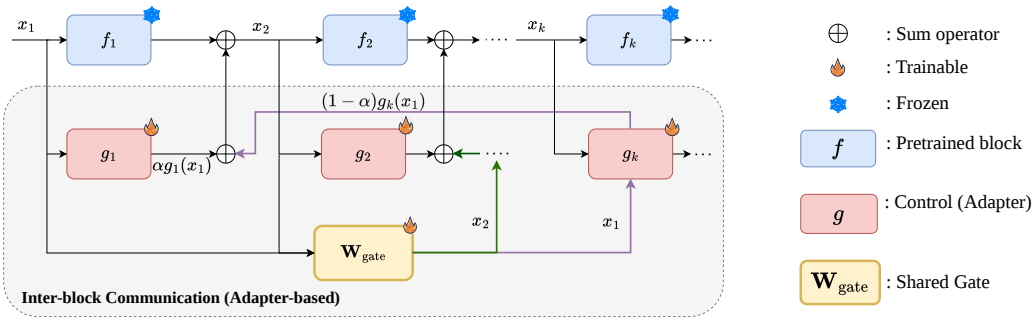


Figure 1: Overview of our Mixture-of-Control (MoC): For an arbitrary input x at the block i, the pretrained block f_i and the local control block g_i process the representation in parallel. A shared gate selects the Top-K control block from all layers, and the selected controller is fused with the local one to form the block control signal. The figure illustrates this mechanism using x_1 and later x_2 as the input under Top-K=1.

4 Methodology

This section introduces Mixture-of-Control (MoC), a generalized state-based fine-tuning framework for efficient cross-block communication. Unlike methods requiring $> 2\times$ forward passes through pretrained blocks, MoC uses lightweight control states to enable information transfer with negligible compute overhead, making it practical for large models and complex architectures.

4.1 Generalized State-Based Tuning: Retaining MoLEx Representational Power

We begin from the following generalized state-based update:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha\, g_t(x_t) + (1-\alpha)\, g_{h_t}(x_t), \tag{7}$$

where W_t denotes the pretrained weight of block t, g_t is its local low-rank control, g_{h_t} is a control chosen from another block h_t, and $\alpha \in [0,1]$ balances local and cross-layer information. Standard state-based tuning is recovered when $h_t = t$, while $h_t \neq t$ enables cross-block communication through lightweight low-rank controls rather than full-block reuse.

The adoption of low-rank controls is motivated by [7], which shows that they efficiently adapt block representations as an alternative to Weight-FT. In constrast to MoLEx [6], which relies on pretrained block reuse under Weight-FT, our method facilitates lightweight cross-block communication while substantially reducing training time and memory overhead. This design choice is further supported by [39], which demonstrates that sufficiently expressive LoRA adapters can approximate arbitrary target functions.

We analyze linear and nonlinear smooth f_t. Generalized state-based fine-tuning retains MoLEx expressivity: it is exact for linear f_t (under mild conditions) and bounded-approximate for nonlinear f_t. We present the linear proof for $f_t(x) = x$; general linear maps follow directly.

Theorem 4.1 (MoLEx Expressivity via State Controls)**.** *Consider the MoLEx update over L blocks defined by*

$$z_{t+1} = z_t + \alpha f_t((W_t + \Delta W_t')z_t) + (1-\alpha) f_{h_t}\left((W_{h_t} + \Delta W_{h_t}')z_t\right), \tag{8}$$

where z_t is the output of the t-th block and the generalized state-based update defined in Equation (7)*. Assuming that $\alpha > \frac{1}{2}$ and $z_1 = x_1$, then there exist controls $g_k : \mathbb{R}^d \to \mathbb{R}^d$ such that: when f_k is linear we have $z_t = x_t$ for all $t \in \{1, \ldots, L+1\}$; moreover, when f_k is smooth nonlinear under small perturbations, i.e., $\forall t$, $\|\Delta W_t'\| < \xi$ ($\xi > 0$ and sufficiently small), then $\mathbb{E}\left[\|z_t - x_t\|^2\right]$ is bounded and vanishes as $\xi \to 0$.*

This theorem indicates that, under mild assumptions, generalized state-based updates can preserve MoLEx-induced representations through state controls. Motivated by this expressivity guarantee, we next develop an adaptive state-based framework that enables inter-block communication.

4.2 Mixture-of-Control (MoC) for State-based Fine-Tuning

As discussed in Section 4.1, inter-block communication can be implemented via low-rank control modules instead of reusable pretrained blocks, substantially reducing computation. We propose Mixture-of-Control (MoC), a lightweight, SMoE-inspired fine-tuning method that captures global information in a depth-adaptive manner. MoC treats the control modules $\{g_k\}_{k=1}^{L}$ as experts and employs a shared gating matrix $\mathbf{W}_{\text{gate}}$ across layers. At each block, the input selects Top-K experts to produce a global control signal; this signal is then combined with local control and added to the pretrained block output before passing to the subsequent block (see Figure 1).

Formally, MoC coordinates local and global control signals adaptively for efficient inter-block information exchange, instantiating Equation (7) as:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha g_t(x_t) + (1-\alpha)\sum_{k=1}^{L} \pi_k(x_t) g_k(x_t), \tag{9}$$

where

$$\pi_k(x_t) = \text{softmax}(\text{TopK}(\mathbf{W}_{\text{gate}} x_t))_k \tag{10}$$

denotes the routing weight for control expert k, L is the number of control sites, and $\mathbf{W}_{\text{gate}}$ is the shared routing matrix. The coefficient α balances local token-specific information with global information injected by routed control experts. As presented in Algorithm 1, MoC adaptively modulates multiple control sites through a shared routing mechanism. In our experiments, we use top-K routing with $K = 1$ and linear low-rank control functions g_k, with ablations on K and control design in Section 5.3. To improve training efficiency for Transformer models, we employ *batch-level routing*, in which token-level routing scores are aggregated using either the mean or the mode. During autoregressive inference, each sample is routed independently based solely on its available prefix. Additional routing details are provided in Section B of Appendix.

MoC extends state-based fine-tuning by combining block-local controls with routed control experts for lightweight inter-block exchange. For Transformers, we study four variants: **MoC (FFN)** routes controls in FFN blocks, and **MoC (ATTN)** in attention blocks. For finer control, **MoC (ATTN+FFN, w/o sharing)** uses separate gates for attention and FFN, while **MoC (ATTN+FFN)** shares a single gate to coordinate both. Unless noted otherwise, **MoC** refers to **MoC (ATTN+FFN)**.

Sharing a gate across attention and FFN follows fine-grained SMoE: granular routing improves expert specialization and utilization without additional cost [40, 41]. In MoC, a shared gate jointly coordinates both pathways for finer adaptation at similar overhead. Overall, MoC enables depth-adaptive Top-K information exchange, increases capacity under a frozen backbone, and remains a lightweight, architecture-agnostic drop-in for standard fine-tuning.

We rigorously evaluate MoC through a theoretical analysis of *stability*, *gradient flow*, and *representation error*, and compare the results to standard Parallel Control.

4.3 Theoretical Analysis

Stability Analysis. We demonstrate that, under the effective routing, MoC exhibits the more stable forward dynamics than the Parallel Control dynamics.

Proposition 4.2 (Stability with Top-K Routing)**.** *Consider a network with L control sites, where each control is a low-rank projection $\Delta W_k = B_k A_k$ with $\|B_k\| \le \beta_k$ and $\|A_k\| \le \gamma$. Let the router select a top-K index set $S_K(x_t)$ with weights $\pi_k(x_t)$ given in Equation* (10)*. Define the effective routed magnitude as $\beta_K(x_t) = \sum_{k\in S_K(x_t)} \pi_k(x_t)\beta_k$.*

Let $\{x_t^c\}_{t=1}^{L+1}$ and $\{x_t\}_{t=1}^{L+1}$ denote the representations produced by Parallel Control and MoC, respectively. Then for each t, $\|x_{t+1}^c\| \le (1 + L_f + \beta_t\gamma)\|x_t^c\|$, whereas MoC with mixing coefficient $\alpha \in [0,1]$ satisfies $\|x_{t+1}\| \le \big(1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma\big)\|x_t\|$, where L_f bounds the Lipschitz contribution of the frozen dynamics. Moreover, if $x_1 = x_1^c$ and the router selects experts such that $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$, then MoC admits a strictly tighter expected one-step growth factor:

$$1 + L_f + \big[\alpha\beta_t + (1-\alpha)\mathbb{E}[\beta_K(x_t)|x_t]\big]\gamma < 1 + L_f + \beta_t\gamma.$$

*Thus, effective routing **yields more stable forward dynamics** by replacing a fixed local control norm with a data-dependent routed norm.*

Algorithm 1: MoC forward pass at block t

Input: Hidden state x_t, block mapping f_t, L block-aligned control experts $\{g_i\}_{i=1}^{L}$, gating function $\mathbf{g}$ over all L experts, sparsity level k, mixing coefficient α.

Output: Updated hidden state x_{t+1}.

$p_{\text{own}} \leftarrow g_t(x_t)$ ▷ layer-dedicated expert

$\boldsymbol{\pi} \leftarrow \mathrm{softmax}\big(\mathbf{g}(x_t)\big)$ ▷ dense routing distribution

$\mathcal{S}_k \leftarrow \mathrm{TopK}\big(\boldsymbol{\pi}, k\big)$ ▷ top-k expert selection

$\pi_i \leftarrow \dfrac{\pi_i}{\sum_{j\in\mathcal{S}_k}\pi_j}, \quad \forall i \in \mathcal{S}_k$ ▷ re-normalized routing weights

$p_{\text{route}} \leftarrow \sum_{i\in\mathcal{S}_k} \pi_i \cdot g_i(x_t)$ ▷ sparse expert aggregation

$g_{\text{moc}}(x_t) \leftarrow \alpha\, p_{\text{own}} + (1-\alpha)\, p_{\text{route}}$ ▷ control signal

$x_{t+1} \leftarrow x_t + f_t(x_t) + g_{\text{moc}}(x_t)$ ▷ update

return x_{t+1}

The key quantity in this result is the routed norm $\beta_K(x_t)$. We enforce $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$ by using a load-balancing loss [36, 37], promoting balanced expert use and better control-expert selection.

Gradient Flow. We show that tighter control of expected forward growth bounds the layerwise Jacobian norms in the backpropagated product, yielding more stable gradients as network depth increases.

Corollary 4.3 (Improved Gradient Flow)**.** *For each time step t, denote by* $J_t = \frac{\partial x_{t+1}}{\partial x_t}$ *and* $J_t^c = \frac{\partial x_{t+1}^c}{\partial x_t^c}$ *the Jacobians of the forward dynamics for the MoC and Parallel Control path, respectively. Under Proposition 4.2's assumptions, MoC admits a tighter expected Jacobian norm upper bound than its Parallel Control counterpart. In particular,* $\forall t$*, we have* $\|J_t^c\| \le M_t^c$ *and* $\|J_t\| \le M_t$*, where* $M_t = 1+L_f\|W_t\|+\alpha\beta_t\gamma+(1-\alpha)\beta_K(x_t)\gamma$ *and* $M_t^c = 1+L_f\|W_t\|+\beta_t\gamma$*. Moreover,* $M_t \le M_t^c, \forall t$*, and consequently,*

$$\prod_{t=1}^{L}\|J_t\| \le \prod_{t=1}^{L} M_t \le \prod_{t=1}^{L} M_t^c, \tag{11}$$

indicating a ***more stable gradient flow*** *under backpropagation in the MoC path.*

Representation Error. Finally, we link adaptive routing to representation quality: by routing and combining multiple control experts, MoC yields a closer approximation to the ideally adaptive control policy than block-local control alone.

Proposition 4.4 (Representation Error Bound)**.** *Consider a network comprising L blocks. Assume* $\Delta W^* x$ *is the optimal control signal that yields task-optimal representations. Denote selected experts exhibit an average error of* $\bar{\epsilon}_K^2 := \mathbb{E}[\|\sum_{k\in S_K(x)}\pi_k(x)\Delta W_k x - \Delta W^* x\|^2|x]$*,* $\bar{\epsilon}_c^2 := \mathbb{E}\left[\|\Delta W_t x - \Delta W^* x\|^2|x\right]$ *and* $\bar{\epsilon}^2 := \mathbb{E}\left[\|g_{moc}(x) - \Delta W^* x\|^2|x\right]$ *are the representation errors of Parallel Control and MoC, respectively. Then the representation error of MoC can be bounded as:*

$$\bar{\epsilon}^2 \le \alpha\bar{\epsilon}_c^2 + (1-\alpha)\bar{\epsilon}_K^2. \tag{12}$$

Furthermore, under Proposition 4.2's assumptions, $\bar{\epsilon}_K^2 < \bar{\epsilon}_c^2$*, which leads to* $\bar{\epsilon}^2 < \bar{\epsilon}_c^2$*.*

These results show a connected benefit of sparse state-control routing: effective routing tightens forward stability bounds, which improves gradient propagation, while the same routed mixture reduces the gap to the ideal representation. The detailed proofs of Theorem 4.1, Proposition 4.2, Corollary 4.3, and Proposition 4.4 are provided in Appendix (see Section A).

5 Experiments

This section describes the experimental setup, reports the main results compared to Weight-FT and State-based baselines, and presents ablation studies. Additional results are provided in the Appendix.

Table 1: Comparison of algorithm performance across commonsense reasoning benchmarks. * marks previously published numbers. Best results are in **bold**.

Model	Method	% Params	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg (↑)	Time (↓) (h)	GPU Mem. (↓) (GB)
LLaMA2-7B	LoRA*	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6	6.1	48.0
	DoRA*	0.85	71.8$_{\uparrow 2.0}$	**83.7**$_{\uparrow 3.8}$	76.0$_{\downarrow 3.5}$	89.1$_{\uparrow 5.5}$	82.6$_{\uparrow 0.0}$	**83.7**$_{\uparrow 3.9}$	68.2$_{\uparrow 3.5}$	**82.4**$_{\uparrow 1.4}$	79.7$_{\uparrow 2.1}$	10.5	80.2
	MoLEx	0.85	69.7$_{\downarrow 0.1}$	81.9$_{\uparrow 2.0}$	78.4$_{\downarrow 1.1}$	91.3$_{\uparrow 7.7}$	81.5$_{\downarrow 1.1}$	81.5$_{\uparrow 1.7}$	65.8$_{\uparrow 1.1}$	80.8$_{\downarrow 0.2}$	78.9$_{\uparrow 1.3}$	9.8	77.4
	FacT_TK	0.51	69.8$_{\uparrow 0.0}$	80.6$_{\uparrow 0.7}$	77.8$_{\downarrow 1.7}$	90.8$_{\uparrow 7.2}$	81.8$_{\downarrow 0.8}$	79.7$_{\downarrow 0.1}$	65.4$_{\uparrow 0.7}$	78.6$_{\downarrow 2.4}$	78.1$_{\uparrow 0.5}$	6.0	48.5
	StructLoRA	0.83	70.9$_{\uparrow 1.1}$	82.0$_{\uparrow 2.1}$	78.6$_{\downarrow 0.9}$	91.6$_{\uparrow 8.0}$	82.1$_{\downarrow 0.5}$	81.6$_{\uparrow 1.8}$	66.6$_{\uparrow 1.9}$	80.6$_{\downarrow 0.4}$	79.3$_{\uparrow 1.7}$	8.5	71.6
	FlexLoRA	0.83	69.5$_{\downarrow 0.3}$	81.2$_{\uparrow 1.3}$	78.1$_{\downarrow 1.4}$	92.9$_{\uparrow 9.3}$	78.7$_{\downarrow 3.9}$	82.8$_{\uparrow 3.0}$	66.5$_{\uparrow 1.8}$	77.0$_{\downarrow 4.0}$	78.3$_{\uparrow 0.7}$	7.0	54.8
	Control*	0.50	**72.3**$_{\uparrow 2.5}$	82.5$_{\uparrow 2.6}$	79.2$_{\downarrow 0.3}$	89.1$_{\uparrow 5.5}$	83.1$_{\uparrow 0.5}$	83.0$_{\uparrow 3.2}$	68.5$_{\uparrow 3.8}$	79.0$_{\downarrow 2.0}$	79.6$_{\uparrow 2.0}$	4.2	42.6
	MoC	0.51	71.8$_{\uparrow 2.0}$	83.2$_{\uparrow 3.3}$	**79.6**$_{\uparrow 0.1}$	**93.3**$_{\uparrow 9.7}$	**84.2**$_{\uparrow 1.6}$	83.6$_{\uparrow 3.8}$	**68.6**$_{\uparrow 3.9}$	80.4$_{\downarrow 0.6}$	**80.5**$_{\uparrow 2.9}$	5.5	47.6
LLaMA3-8B	LoRA*	0.71	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8	7.5	59.1
	DoRA*	0.71	74.6$_{\uparrow 3.8}$	89.3$_{\uparrow 4.1}$	79.9$_{\uparrow 0.0}$	95.5$_{\uparrow 3.8}$	85.6$_{\uparrow 1.3}$	90.5$_{\uparrow 6.3}$	80.4$_{\uparrow 9.2}$	85.8$_{\uparrow 6.8}$	85.2$_{\uparrow 4.4}$	10.5	69.1
	MoLEx	0.72	70.1$_{\downarrow 0.7}$	86.2$_{\uparrow 1.0}$	80.4$_{\uparrow 0.5}$	93.2$_{\uparrow 1.5}$	84.1$_{\downarrow 0.2}$	84.8$_{\uparrow 0.6}$	71.3$_{\uparrow 0.1}$	82.2$_{\uparrow 3.2}$	81.5$_{\uparrow 0.7}$	11.0	94.9
	FacT_TK	0.43	70.7$_{\downarrow 0.1}$	86.0$_{\uparrow 0.8}$	79.5$_{\downarrow 0.4}$	94.2$_{\uparrow 2.5}$	84.2$_{\downarrow 0.1}$	86.5$_{\uparrow 2.3}$	73.3$_{\uparrow 2.1}$	81.8$_{\uparrow 2.8}$	82.0$_{\uparrow 1.2}$	7.7	57.6
	StructLoRA	0.71	70.9$_{\uparrow 0.1}$	86.6$_{\uparrow 1.4}$	79.5$_{\downarrow 0.4}$	93.9$_{\uparrow 2.2}$	84.0$_{\downarrow 0.3}$	87.0$_{\uparrow 2.8}$	73.9$_{\uparrow 2.7}$	82.6$_{\uparrow 3.6}$	82.3$_{\uparrow 1.5}$	9.8	75.4
	FlexLoRA	0.71	73.4$_{\uparrow 2.6}$	88.5$_{\uparrow 3.3}$	78.6$_{\downarrow 1.3}$	95.3$_{\uparrow 3.6}$	84.4$_{\uparrow 0.1}$	90.2$_{\uparrow 6.0}$	79.2$_{\uparrow 8.0}$	85.6$_{\uparrow 6.6}$	84.4$_{\uparrow 3.6}$	8.6	62.4
	Control*	0.42	74.1$_{\uparrow 3.3}$	87.8$_{\uparrow 2.6}$	80.7$_{\uparrow 0.8}$	95.5$_{\uparrow 3.8}$	86.0$_{\uparrow 1.7}$	90.8$_{\uparrow 6.6}$	80.0$_{\uparrow 8.8}$	87.8$_{\uparrow 8.8}$	85.3$_{\uparrow 4.5}$	4.4	54.0
	MoC	0.43	**75.5**$_{\uparrow 4.7}$	**89.5**$_{\uparrow 4.3}$	**82.0**$_{\uparrow 2.1}$	**96.6**$_{\uparrow 4.9}$	**88.8**$_{\uparrow 4.5}$	**91.5**$_{\uparrow 7.3}$	**81.6**$_{\uparrow 10.4}$	**88.0**$_{\uparrow 9.0}$	**86.7**$_{\uparrow 5.9}$	6.5	56.7
Mistralv3-7B	LoRA	0.78	68.7	83.5	77.2	90.3	82.5	85.7	73.4	80.5	80.2	6.5	58.4
	DoRA	0.79	70.3$_{\uparrow 1.5}$	84.5$_{\uparrow 1.0}$	77.8$_{\uparrow 0.6}$	89.9$_{\downarrow 0.4}$	83.1$_{\uparrow 0.5}$	87.1$_{\uparrow 1.4}$	72.2$_{\downarrow 1.2}$	82.4$_{\uparrow 1.9}$	80.9$_{\uparrow 0.7}$	10.1	70.4
	MoLEx	0.80	71.4$_{\uparrow 2.7}$	85.4$_{\uparrow 2.0}$	78.2$_{\uparrow 1.0}$	90.3$_{\downarrow 0.0}$	83.5$_{\uparrow 1.0}$	84.1$_{\downarrow 1.6}$	75.0$_{\uparrow 1.5}$	82.7$_{\uparrow 2.2}$	81.3$_{\uparrow 1.1}$	10.5	82.3
	FacT_TK	0.48	70.5$_{\uparrow 1.8}$	86.4$_{\uparrow 2.9}$	80.3$_{\uparrow 3.1}$	94.1$_{\uparrow 3.8}$	84.2$_{\uparrow 1.7}$	83.9$_{\downarrow 1.8}$	71.0$_{\downarrow 2.4}$	84.4$_{\uparrow 3.9}$	81.9$_{\uparrow 1.7}$	6.3	56.0
	StructLoRA	0.78	70.0$_{\uparrow 1.3}$	83.2$_{\downarrow 0.3}$	77.4$_{\uparrow 0.2}$	89.9$_{\downarrow 0.4}$	81.7$_{\downarrow 0.8}$	87.2$_{\uparrow 1.5}$	74.8$_{\uparrow 1.4}$	81.4$_{\uparrow 0.9}$	80.7$_{\uparrow 0.5}$	9.2	70.1
	FlexLoRA	0.78	73.9$_{\uparrow 5.2}$	88.4$_{\uparrow 4.9}$	79.2$_{\uparrow 2.0}$	**95.9**$_{\uparrow 5.6}$	86.0$_{\uparrow 3.5}$	**90.2**$_{\uparrow 4.5}$	78.8$_{\uparrow 5.4}$	86.8$_{\uparrow 6.3}$	84.9$_{\uparrow 4.7}$	7.5	61.3
	Control	0.46	70.6$_{\uparrow 1.8}$	86.9$_{\uparrow 3.5}$	79.0$_{\uparrow 1.8}$	94.2$_{\uparrow 3.9}$	84.8$_{\uparrow 2.2}$	87.9$_{\uparrow 2.2}$	76.5$_{\uparrow 3.0}$	84.8$_{\uparrow 4.3}$	83.1$_{\uparrow 2.9}$	4.3	53.2
	MoC	0.48	**74.2**$_{\uparrow 5.5}$	**88.9**$_{\uparrow 5.4}$	**81.0**$_{\uparrow 3.8}$	95.6$_{\uparrow 5.3}$	**87.9**$_{\uparrow 5.4}$	88.9$_{\uparrow 3.2}$	**79.5**$_{\uparrow 6.1}$	**87.8**$_{\uparrow 7.3}$	**85.5**$_{\uparrow 5.3}$	5.5	55.6
Qwen2.5-14B	LoRA	0.63	73.4	90.4	82.2	96.0	88.6	94.7	86.1	90.6	87.7	9.4	62.1
	DoRA	0.64	75.1$_{\uparrow 1.7}$	90.1$_{\downarrow 0.3}$	82.0$_{\downarrow 0.2}$	96.3$_{\uparrow 0.3}$	89.4$_{\uparrow 0.8}$	95.0$_{\uparrow 0.3}$	88.2$_{\uparrow 2.1}$	93.0$_{\uparrow 2.4}$	88.6$_{\uparrow 0.9}$	14.3	94.7
	MoLEx	0.64	73.5$_{\uparrow 0.1}$	90.5$_{\uparrow 0.1}$	81.9$_{\downarrow 0.3}$	95.5$_{\downarrow 0.5}$	88.2$_{\downarrow 0.4}$	94.4$_{\downarrow 0.3}$	85.8$_{\downarrow 0.3}$	92.2$_{\uparrow 1.6}$	87.8$_{\uparrow 0.1}$	20.3	124.8
	FacT_TK	0.43	74.7$_{\uparrow 1.3}$	90.1$_{\downarrow 0.3}$	82.3$_{\uparrow 0.1}$	96.2$_{\uparrow 0.2}$	89.6$_{\uparrow 1.0}$	94.7$_{\uparrow 0.0}$	88.1$_{\uparrow 2.0}$	92.4$_{\uparrow 1.8}$	88.5$_{\uparrow 0.8}$	8.0	61.5
	StructLoRA	0.63	74.8$_{\uparrow 1.4}$	89.5$_{\downarrow 0.9}$	81.0$_{\downarrow 1.2}$	95.8$_{\downarrow 0.2}$	89.3$_{\uparrow 0.7}$	94.7$_{\uparrow 0.0}$	87.4$_{\uparrow 1.3}$	92.4$_{\uparrow 1.8}$	88.1$_{\uparrow 0.4}$	13.5	77.3
	FlexLoRA	0.63	74.2$_{\uparrow 0.8}$	91.4$_{\uparrow 1.0}$	82.0$_{\downarrow 0.2}$	96.4$_{\uparrow 0.4}$	87.7$_{\downarrow 0.9}$	**96.7**$_{\uparrow 2.0}$	90.0$_{\uparrow 3.9}$	95.4$_{\uparrow 4.8}$	89.2$_{\uparrow 1.5}$	10.5	67.4
	Control	0.43	72.8$_{\downarrow 0.6}$	88.8$_{\downarrow 1.6}$	81.3$_{\downarrow 0.9}$	94.6$_{\downarrow 1.4}$	87.5$_{\downarrow 1.1}$	94.4$_{\downarrow 0.3}$	86.9$_{\uparrow 0.8}$	91.2$_{\uparrow 0.6}$	87.2$_{\downarrow 0.5}$	6.5	58.7
	MoC	0.43	**75.5**$_{\uparrow 2.1}$	**91.8**$_{\uparrow 1.4}$	**82.4**$_{\uparrow 0.2}$	**96.8**$_{\uparrow 0.8}$	**91.0**$_{\uparrow 2.4}$	96.6$_{\uparrow 1.9}$	**90.2**$_{\uparrow 4.1}$	**96.4**$_{\uparrow 5.8}$	**90.1**$_{\uparrow 2.4}$	8.0	61.3

5.1 Experimental Settings

Datasets. We conduct experiments on diverse language and vision benchmarks. For natural language generation (NLG), we use 8 commonsense reasoning datasets: BoolQ [42], PiQA [43], Social IQa [44], HellaSwag [45], WinoGrande [46], ARC-E/ARC-C [47], and OpenBookQA [48]. For natural language understanding (NLU), we use GLUE [49], which includes datasets such as SST-2, MRPC, CoLA, QNLI, RTE, and STS-B. For vision, we use VTAB-1K [50], which contains 19 datasets spanning Natural, Specialized, and Structured domains. These benchmarks provide a comprehensive evaluation across language and vision tasks.

Backbone models. For NLG, we use LLaMA2-7B [51], LLaMA3-8B [52], Mistral-7B Instruct [53], and Qwen2.5-14B Instruct [54]. For NLU, we evaluate RoBERTa and DeBERTa in both base and large variants. For vision tasks, we adopt ViT-Tiny and ViT-Base as standard backbones.

Baselines. Across domains, we compare against some recent, representative adaptation methods, including LoRA [55], DoRA [56], MoLEx [6], FacT_TK [24], StructLoRA [25], FlexLoRA [26], and the recent Parallel Control [7] (abbreviated as Control).

Evaluation Metrics and Analysis. For NLG and vision tasks, we evaluate performance using accuracy for eight commonsense reasoning datasets and 19 visual datasets, respectively. For NLU tasks, we follow standard evaluation protocols and report task-specific metrics: Matthew's correlation coefficient for CoLA, Pearson correlation for STS-B, and accuracy for all remaining tasks.

5.2 Main Results

We use LoRA as the common reference baseline; superscripts ↑ and ↓ denote improvement and degradation relative to LoRA, respectively.

NLG Tasks. We evaluate MoC on NLG tasks (Table 1). MoC consistently outperforms prior methods across backbones from LLaMA2-7B to Qwen2.5-14B, beating LoRA by 2.4%–6.0% and

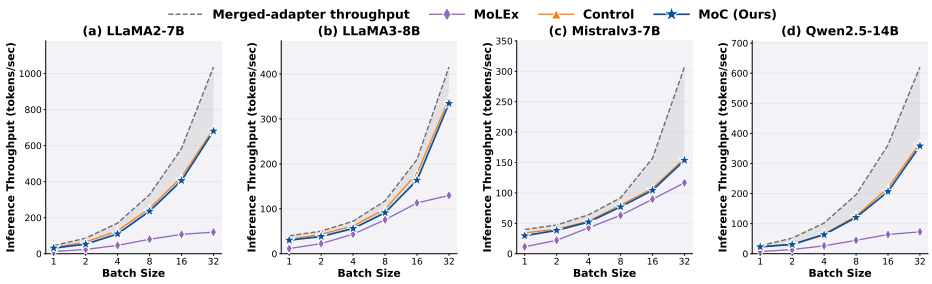


Figure 2: Inference Throughput of MoC compared to prior methods across LLMs and batch size.

Table 2: GLUE results with RoBERTa-Base/Large backbones. Values are medians of five runs (different seeds); * marks previously published numbers. Best results are in **bold**.

Model	Method	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg (↑)
RoBERTa-Base	LoRA*	95.1	89.7	63.4	93.3	78.4	91.5	85.2
	DoRA*	95.0↓0.1	89.7↑0.0	64.9↑1.5	92.9↓0.4	79.2↑0.8	91.3↓0.2	85.5↑0.3
	MoLEx*	95.4↑0.3	89.8↑0.1	64.8↑1.4	93.2↓0.1	77.3↓1.1	91.0↓0.5	85.3↑0.1
	Control*	95.3↑0.2	89.3↓0.4	65.3↑1.9	93.1↓0.2	76.9↓1.5	90.6↓0.9	85.1↓0.1
	MoC	**95.6**↑0.5	**90.7**↑1.0	**66.7**↑3.3	**93.4**↑0.1	**81.9**↑3.5	**91.5**↑0.0	**86.7**↑1.5
RoBERTa-Large	LoRA*	96.2	90.2	68.2	94.8	85.2	92.3	87.8
	DoRA*	96.6↑0.4	90.9↑0.7	68.8↑0.6	95.2↑0.4	84.7↓0.5	92.5↑0.2	88.1↑0.3
	MoLEx*	96.4↑0.2	91.4↑1.2	68.2↑0.0	94.8↑0.0	87.1↑1.9	92.0↓0.3	88.3↑0.5
	Control*	96.7↑0.5	90.9↑0.7	69.9↑1.7	95.2↑0.4	86.3↑1.1	92.5↑0.2	88.6↑0.8
	MoC	**97.2**↑1.0	**92.6**↑2.4	**71.4**↑3.2	**95.2**↑0.4	**89.9**↑4.7	**92.5**↑0.2	**89.8**↑2.0

the strongest baseline by 0.6%–1.4% with comparable compute. While MoLEx also enables cross-block communication, it costs up to $> 2\times$ time and memory on Qwen2.5-14B; MoC achieves better performance with superior scalability. Beyond accuracy, Figure 2 compares the inference throughput of MoC with that of representative baselines offering competitive efficiency–performance trade-offs across model scales and batch sizes. While state-based methods remain less efficient than fully merged Weight-FT, MoC consistently outperforms MoLEx in throughput–even after merging–improving the accuracy–efficiency trade-off and narrowing the gap to mergeable fine-tuning. Besides, a detailed efficiency comparison between MoC and representative baselines is provided in Section B.3.

NLU Tasks. Next, we evaluate the MoC against previous approaches on the GLUE benchmark in Table 2. MoC consistently outperforms competing baselines across all six datasets, achieving average gains of 1.5% and 2.0% over LoRA for RoBERTa-Base and RoBERTa-Large, respectively, and outperforming the strongest competing baseline by 1.2%. Together, these results highlight the effectiveness of MoC across both NLG and NLU settings. We further report DeBERTa results in Appendix (see Table 10 for details).

Vision Tasks. To test whether MoC's gains extend beyond text, we evaluate it on VTAB-1K with ViT-Tiny and ViT-Base. As shown in Table 3, MoC achieves the best average accuracy across 19 datasets spanning Natural, Specialized, and Structured domains, improving over LoRA by 1.3% and 1.4%, and over the strongest baseline by 1.3% and 1.2% on ViT-Tiny and ViT-Base, respectively. MoC also shows competitive training time and memory usage compared with LoRA, DoRA, and MoLEx, highlighting its advantages in performance, time, and memory efficiency. Detailed per-dataset results and plots are provided in the Appendix; see Figure 4 and Table 11.

Overall, these results show that MoC provides a unified and efficient adaptation strategy across language and vision tasks.

5.3 Ablation Studies

TopK Selection. We analyze Top-K selection across LLM backbones and report training time and GPU memory on Mistralv3-7B to evaluate scalability. As shown in Figure 3(b), MoC performs best with two activated experts, followed by eight. In MoC, Top-$K = 1$ pairs the block-local controller with one routed controller from other blocks, similar to Top-2 routing in sparse MoE and offering a strong accuracy–efficiency trade-off [37, 57, 58]. Unlike standard MoE, MoC selects experts across blocks rather than within a layer, so increasing Top-K changes both control capacity and inter-block information exchange.
We also test $K \in \{0, 1, 3, 7\}$ (correspond to the number of activated experts $\in \{1, 2, 4, 8\}$), where

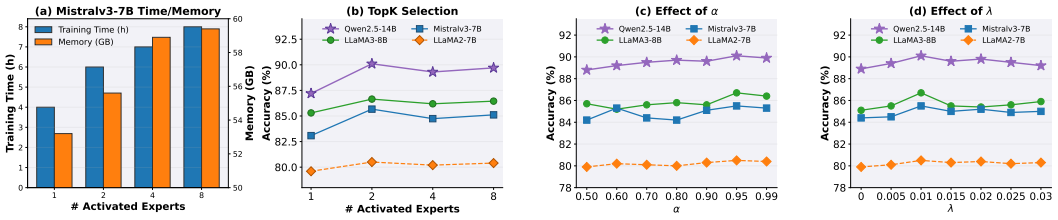


Figure 3: Ablation analysis of MoC under different configurations. (a) Training-time and GPU-memory overhead as the number of activated experts varies, reported on Mistralv3-7B. (b–d) Performance analysis of MoC across different LLM backbones, including (b) TopK Selection, (c) the effect of the coefficient α, and (d) the effect of the coefficient λ.

Table 3: Comparison on VTAB-style benchmarks.

Model	Method	Natural (↑)	Specialized (↑)	Structured (↑)	Avg (↑)	Time (m) (↓)	GPU Mem. (GB) (↓)
ViT-Tiny	LoRA	67.5	84.6	49.9	63.7	1.5	5.4
	DoRA	67.0$_{\downarrow 0.5}$	84.2$_{\downarrow 0.4}$	49.9$_{\uparrow 0.0}$	63.4$_{\downarrow 0.3}$	1.9$_{\downarrow 0.4}$	9.2$_{\downarrow 3.8}$
	MoLEx	68.1$_{\uparrow 0.6}$	84.4$_{\downarrow 0.2}$	49.4$_{\downarrow 0.5}$	63.6$_{\downarrow 0.1}$	2.6$_{\downarrow 1.1}$	11.6$_{\downarrow 6.2}$
	Control	65.7$_{\downarrow 1.8}$	84.4$_{\downarrow 0.2}$	47.1$_{\downarrow 2.8}$	61.8$_{\downarrow 1.9}$	1.0$_{\uparrow 0.5}$	3.3$_{\uparrow 2.1}$
	MoC	**70.0**$_{\uparrow 2.5}$	**85.9**$_{\uparrow 1.3}$	**50.2**$_{\uparrow 0.3}$	**65.0**$_{\uparrow 1.3}$	1.4$_{\uparrow 0.1}$	3.6$_{\uparrow 1.8}$
ViT-Base	LoRA	77.9	85.9	48.1	67.0	4.5	21.6
	DoRA	78.1$_{\uparrow 0.2}$	85.1$_{\downarrow 0.8}$	47.1$_{\downarrow 0.9}$	66.6$_{\downarrow 0.4}$	8.7$_{\downarrow 4.2}$	36.6$_{\downarrow 15.0}$
	MoLEx	78.1$_{\uparrow 0.1}$	86.9$_{\uparrow 1.0}$	47.5$_{\downarrow 0.6}$	67.1$_{\uparrow 0.1}$	10.6$_{\downarrow 6.1}$	39.9$_{\downarrow 18.3}$
	Control	77.6$_{\downarrow 0.3}$	86.5$_{\uparrow 0.6}$	48.5$_{\uparrow 0.4}$	67.2$_{\uparrow 0.2}$	3.9$_{\uparrow 0.6}$	13.8$_{\uparrow 7.8}$
	MoC	**79.0**$_{\uparrow 1.1}$	**87.6**$_{\uparrow 1.8}$	**49.4**$_{\uparrow 1.4}$	**68.4**$_{\uparrow 1.4}$	4.2$_{\uparrow 0.3}$	13.8$_{\uparrow 7.8}$

$K = 0$ uses only the local controller (*Control baseline*); see Figure 3(a). Training time and memory increase moderately with Top-K and remain well below MoLEx in Table 1, indicating efficient scaling. Routing visualizations further show adaptive use of local and global information, supporting cross-block control (see Figures 5 to 13 in Appendix).

Mixing Coefficient α. We study the sensitivity of MoC to the mixing coefficient α. Motivated by Theorem 4.1, we vary $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ while fixing $\lambda = 0.01$. As shown in Figure 3 (c), MoC achieves the best performance at $\alpha = 0.95$ across different LLMs. It also remains competitive or superior to the baselines in Table 1 over a wide range of α values, indicating robustness to α while benefiting from the expressive inter-block communication characterized in Theorem 4.1.

Load-Balancing Coefficient λ. We examine the MoC's sensitivity to the load-balancing coefficient λ by fixing $\alpha = 0.95$ and varying $\lambda \in \{0, 0.005, 0.01, \ldots, 0.03\}$, where $\lambda = 0$ removes load balancing. As shown in Figure 3 (d), load balancing generally improves performance, indicating better expert utilization and routing and closer alignment with the conditions in Proposition 4.2, Corollary 4.3, and Proposition 4.4. Performance peaks at $\lambda = 0.01$, suggesting a small coefficient suffices, while larger values can hinder the main objective.

MoC Variants. We compare MoC variants that control attention, FFN, or both (Table 4); FFN control consistently beats attention-only, and MoC (FFN+ATTN) with a shared gate performs best, matching the view that attention routes information while FFNs drive nonlinear semantic transformation [59, 60]. Shared gating over both modules enables richer cross-layer control and finer expert selection, strengthening cross-layer communication in line with fine-grained SMoE principles (Section 4.2).

Table 4: Comparison of different MoC variants on LLaMA3-8B and Mistralv3-7B.

MoC Variant	LLaMA3-8B (↑)	Mistralv3-7B (↑)
ATTN	85.9	84.9
FFN	86.2	85.1
ATTN+FFN w/o sharing	86.4	84.9
ATTN+FFN	**86.7**	**85.5**

Table 5: Different control activations on VTAB benchmarks using ViT-Base backbone.

Activation	Natural (↑)	Specialized (↑)	Structured (↑)	Avg (↑)
Identity	79.0	**87.6**	**49.4**	**68.4**
ReLU	**79.1**	86.5	48.9	67.9
GeLU	78.7	86.8	**49.4**	68.1
SiLU	**79.1**	86.6	48.8	67.9

Routing Design and Expert-Pool Size. We investigate the individual contributions of cross-layer communication, adaptive routing, and expert-pool size on Mistralv3-7B-Instruct. As shown in Table 6,

Table 6: Ablation of different cross-layer communication designs on Mistralv3-7B-Instruct.

Method	Avg. Acc. (↑)	GPU Mem. (↓)
Control	83.1	53.2
Uniform Control	84.3	53.8
Fixed Control	84.2	55.0
MoC w/o lb	84.4	55.4
MoC	**85.5**	55.6

Table 7: Effect of expert-pool size on Mistralv3-7B-Instruct. More groups yield smaller routing pools.

# Groups	# Experts	Avg. Acc. (↑)	Time (↓)	GPU Mem. (↓)
1	**64**	**85.5**	6.1	55.6
2	32	85.1	6.1	55.6
4	16	85.0	6.1	55.4
8	8	84.8	6	55.4
16	4	84.8	6	55.4

all communication variants outperform the local Control baseline. Uniform and Fixed Control achieve accuracies of 84.3% and 84.2%, showing that cross-layer alignment alone is beneficial, while adaptive routing without load balancing reaches 84.4%. Adding load balancing further improves MoC to 85.5% (+1.1 points), supporting balanced input-dependent expert selection and the motivation of Proposition 4.2 and Corollary 4.3, with only modest memory overhead. Table 7 further shows that reducing the expert pool from 64 to 4 lowers accuracy from 85.5% to 84.8%, while training time and memory remain nearly unchanged. This suggests that MoC benefits from broad access to cross-layer controls rather than additional computation, highlighting the value of flexible global routing.

Control Types. We study MoC with alternative controller designs. In our main setup, each controller is a linear low-rank projection, offering a strong performance–efficiency trade-off (Section 4.1). More generally, we define $g_i(x) = \sigma_2(B_i\sigma_1(A_i x))$, where σ_1 and σ_2 can be linear or nonlinear. For simplicity, we set σ_2 to the identity and consider $\sigma_1 \in \{\text{Identity}, \text{ReLU}, \text{GeLU}, \text{SiLU}\}$, yielding a generalized family of nonlinear adapters. As shown in Table 5, MoC extends naturally to these variants, and nonlinear controllers are competitive across tasks. Nonetheless, the linear controller remains highly effective while preserving simplicity and efficiency. We leave theoretical analysis of nonlinear controllers to future work.

6 Conclusion

In this paper, we identify two limitations of efficient fine-tuning: (i) state-based methods restrict adaptation to per-block control, limiting multi-block coordination; and (ii) existing cross-block communication methods often incur substantial compute and memory overhead. We introduce Mixture-of-Control (MoC), a lightweight framework that treats block-wise control states as experts and adaptively combines local and global signals through sparse cross-block routing. MoC enables efficient multi-block coordination while preserving the memory benefits of state-based fine-tuning. Experiments across language and vision domains with diverse backbones show that MoC consistently outperforms prior baselines with minimal overhead and favorable inference throughput. Taken together, these results highlight control-driven cross-block communication as a promising direction for scalable Transformer adaptation. Future work includes exploring more expressive control architectures and further narrowing the inference-efficiency gap to mergeable fine-tuning methods.

Limitations.

MoC is subject to several limitations. First, unlike mergeable weight-based methods such as LoRA, its state-dependent control and routing modules remain active at inference, introducing overhead that may reduce throughput, especially with many control sites. Developing more efficient routing and control mechanisms is therefore important for practical deployment. Second, our theory currently focuses on linear low-rank controllers. Although nonlinear variants can further improve performance (see Table 5), establishing comparable expressivity, approximation, and stability guarantees remains open. Third, the mixing coefficient α and load-balancing coefficient λ are currently selected manually. While these choices are guided by our theory and standard MoE practice and remain robust across the tested ranges, dynamically or jointly learning them could improve adaptation across models and tasks. Finally, extending cross-layer communication to architectures such as looped Transformers [61, 62] raises broader research questions about representation compatibility, depth-wise learning dynamics, and input-dependent communication.

Acknowledgments and Disclosure of Funding

Funding and Support. This research was conducted in part at Qualcomm AI Research, an initiative of Qualcomm Technologies, Inc., with research resources provided by the company.

Competing Interests. All authors were affiliated with Qualcomm Technologies, Inc. during the conduct of this research. The authors declare no other competing interests.

References

[1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[2] Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

[3] Chi Zhang, Jingpu Cheng, and Qianxiao Li. An optimal control view of lora and binary controller design for vision transformers. In *European Conference on Computer Vision*, pages 144–160. Springer, 2024.

[4] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

[5] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Rachel SY Teo and Tan M Nguyen. Molex: Mixture of layer experts for finetuning with sparse upcycling. *arXiv preprint arXiv:2503.11144*, 2025.

[7] Chi Zhang, REN Lianhai, Jingpu Cheng, and Qianxiao Li. From weight-based to state-based fine-tuning: Further memory reduction on lora with parallel control. In *Forty-second International Conference on Machine Learning*, 2025.

[8] Chi Zhang, Cheng Jingpu, Yanyu Xu, and Qianxiao Li. Parameter-efficient fine-tuning with controls. In *Forty-first International Conference on Machine Learning*, 2024.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[11] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014. doi: 10.1109/CVPR.2014.222.

[12] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018. URL https://arxiv.org/abs/1610.01644.

[13] Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts, 2023. URL https://arxiv.org/abs/2210.11466.

[14] Gal Kaplun, Andrey Gurevich, Tal Swisa, Mazor David, Shai Shalev-Shwartz, and Eran Malach. Less is more: Selective layer finetuning with subtuning. *arXiv preprint arXiv:2302.06354*, 2023.

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. URL https://arxiv.org/abs/1902.00751.

[16] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers, 2021. URL https://arxiv.org/abs/2106.04647.

[17] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

[18] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

[19] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021. URL https://arxiv.org/abs/2101.00190.

[20] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models, 2023. URL https://arxiv.org/abs/2301.12314.

[21] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

[22] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning, 2023. URL https://arxiv.org/abs/2108.06098.

[23] Zheda Mai, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Quang-Huy Nguyen, Li Zhang, and Wei-Lun Chao. Lessons and insights from a unifying study of parameter-efficient fine-tuning (peft) in visual recognition. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14845–14857. IEEE, 2025.

[24] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 1060–1068, 2023.

[25] Xi Xiao, Chenrui Ma, Yunbei Zhang, Chen Liu, Zhuxuanzi Wang, Yanshu Li, Lin Zhao, Guosheng Hu, Tianyang Wang, and Hao Xu. Not all directions matter: Towards structured and task-aware low-rank model adaptation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2132–2154, 2026.

[26] Muqing Liu, Chongjie Si, and Yuheng Jia. Flexlora: Entropy-guided flexible low-rank adaptation. *arXiv preprint arXiv:2601.22905*, 2026.

[27] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

[28] Qianxiao Li, Long Chen, Cheng Tai, and Weinan E. Maximum principle based algorithms for deep learning, 2018. URL https://arxiv.org/abs/1710.09513.

[29] Paulo Tabuada and Bahman Gharesifard. Universal approximation power of deep residual neural networks via nonlinear control theory, 2024. URL https://arxiv.org/abs/2007.06007.

[30] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations, 2020. URL https://arxiv.org/abs/1710.10121.

[31] Tam Nguyen, César A. Uribe, Tan M. Nguyen, and Richard G. Baraniuk. Pidformer: Transformer meets control theory, 2024. URL https://arxiv.org/abs/2402.15989.

[32] Martin Benning, Elena Celledoni, Matthias J. Ehrhardt, Brynjulf Owren, and Carola-Bibiane Schönlieb. Deep learning as optimal control problems: models and numerical methods, 2019. URL https://arxiv.org/abs/1904.05657.

[33] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328, 2021.

[34] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[35] Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River, 2002.

[36] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022.

[37] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[38] Duc Anh Nguyen, Huu Binh Ta, Nhuan Le Duc, Tan Minh Nguyen, and Toan Tran. Selective sinkhorn routing for improved sparse mixture of experts. *arXiv preprint arXiv:2511.08972*, 2025.

[39] Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. *arXiv preprint arXiv:2310.17513*, 2023.

[40] Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. *arXiv preprint arXiv:2402.07871*, 2024.

[41] Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447, 2025.

[42] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[43] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[44] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

[45] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[46] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[47] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[48] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[49] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355, 2018.

[50] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[51] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[52] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[53] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[54] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

[55] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL `https://arxiv.org/abs/2106.09685`.

[56] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024. URL `https://arxiv.org/abs/2402.09353`.

[57] Rachel Teo and Tan Minh Nguyen. MomentumSMoe: Integrating momentum into sparse mixture of experts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=y929esCZNJ`.

[58] Yanqi Zhou, Nan Du, Yanping Huang, Daiyi Peng, Chang Lan, Da Huang, Siamak Shakeri, David So, Andrew M Dai, Yifeng Lu, et al. Brainformers: Trading simplicity for efficiency. In *International conference on machine learning*, pages 42531–42542. PMLR, 2023.

[59] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[60] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

[61] Kevin Xu and Issei Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. *arXiv preprint arXiv:2410.01405*, 2024.

[62] Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. Loopformer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. *arXiv preprint arXiv:2602.11451*, 2026.

[63] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

[64] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[65] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Mixture-of-Control: State-Aware Fine-Tuning for Transformer-based Models
Supplementary Material

A Proofs for Theoretical Results

A.1 Theorem 4.1

Proof. For simplicity, in the linear case, we take f_t to be an identity-like mapping. Thus, MoLEx update is simplified into:

$$z_{t+1} = z_t + [\alpha(W_t + \Delta W'_t) + (1-\alpha)(W_{h_t} + \Delta W'_{h_t})]z_t \tag{13}$$
$$= z_t + W_t z_t + [\alpha(\Delta W'_t) + (1-\alpha)(W_{h_t} - W_t + \Delta W'_{h_t})]z_t \tag{14}$$

Denote $M_t = \alpha(\Delta W'_t) + (1-\alpha)(W_{h_t} - W_t + \Delta W'_{h_t})$, for $t = \overline{1, L}$. Similarly, the generalized state-based update is formulated as follows:

$$\begin{aligned} x_{t+1} &= x_t + W_t x_t + \alpha \Delta W_t x_t + (1-\alpha)\Delta W_{h_t} x_t \\ &= x_t + W_t x_t + (\alpha \Delta W_t + (1-\alpha)\Delta W_{h_t})x_t \end{aligned}$$

Consequently, under the initial condition $x_1 = z_1$, the equivalence between the two systems can be demonstrated by the following set of equations:

$$\begin{cases} \alpha \Delta W_1 + (1-\alpha)\Delta W_{h_1} = M_1 \\ \alpha \Delta W_2 + (1-\alpha)\Delta W_{h_2} = M_2 \\ ... \\ \alpha \Delta W_L + (1-\alpha)\Delta W_{h_L} = M_L. \end{cases}$$

This admits the following matrix representation:

$$\begin{bmatrix} \alpha & 0 & 0 & \dots & 1-\alpha & 0 & \dots & 0 \\ 0 & \alpha & 0 & \dots & 0 & 1-\alpha & \dots & 0 \\ 0 & 0 & \alpha & \dots & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & 1-\alpha & 0 & 0 & \dots & \alpha \end{bmatrix} \begin{bmatrix} \Delta W_1 \\ \Delta W_2 \\ \Delta W_3 \\ \Delta W_4 \\ \Delta W_5 \\ \Delta W_6 \\ \vdots \\ \Delta W_L \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ \vdots \\ M_L \end{bmatrix}.$$

Denote

$$A = \begin{bmatrix} \alpha & 0 & 0 & \dots & 1-\alpha & 0 & \dots & 0 \\ 0 & \alpha & 0 & \dots & 0 & 1-\alpha & \dots & 0 \\ 0 & 0 & \alpha & \dots & 0 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & 1-\alpha & 0 & 0 & \dots & \alpha \end{bmatrix},$$

where $A = \alpha I + (1-\alpha)B$, where $I \in \mathbb{R}^{Ld \times Ld}$ is the identity matrix, and $B \in \mathbb{R}^{Ld \times Ld}$ is a matrix with exactly Ld entries equal to 1 (with positions determined by W_{h_t}), and all other entries equal to 0.

Next, we prove that $\alpha > \frac{1}{2}$ ensures $\det(A) \neq 0$. Suppose, for contradiction, that there exists a nonzero vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{Ld} \end{bmatrix} \neq \mathbf{0},$$

such that

$$A\mathbf{x} = 0.$$

Then,

$$(\alpha I + (1-\alpha)B)\mathbf{x} = 0 \quad \iff \quad B\mathbf{x} = -\frac{\alpha}{1-\alpha}\mathbf{x}.$$

Writing this component-wise, we obtain

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_{Ld}} \end{bmatrix} = \frac{\alpha}{1-\alpha} \begin{bmatrix} -x_1 \\ -x_2 \\ \vdots \\ -x_{Ld} \end{bmatrix},$$

where $\{i_1, i_2, \dots, i_{Ld}\} \subseteq \{1, 2, \dots, Ld\}$.

Taking the maximum absolute value on both sides yields

$$\max_{1 \leq j \leq Ld} |x_{i_j}| = \frac{\alpha}{1-\alpha} \max_{1 \leq j \leq Ld} |x_j|.$$

Since $\alpha > \frac{1}{2}$ implies $\frac{\alpha}{1-\alpha} > 1$, we obtain

$$\max_{1 \leq j \leq Ld} |x_{i_j}| > \max_{1 \leq j \leq Ld} |x_j|,$$

which is a contradiction. Therefore, no such nonzero $\mathbf{x}$ exists, and hence $\det(A) \neq 0$. Consequently, there exists a *unique* solution $\{\Delta W_k\}_{k=1}^{L}$ that establishes representational equivalence between the generalized state-based formulation and MoLEx.

In the nonlinear case, we show that the equivalence holds approximately when the adapters $\{\Delta W'_k\}_{k=1}^{L}$ are low-norm perturbations, using local linearization around the frozen pretrained path. Assume that each activation function f_k ($k = 1, \dots, L$) is twice continuously differentiable, with

- Lipschitz constant $L_f < \infty$ (uniform over k): $\|f_k(a) - f_k(b)\| \leq L_f \|a - b\|$ for all a, b,
- Bounded Hessian norm $H < \infty$ (uniform over k): $\|\nabla^2 f_k(z)\| \leq H$ for all z (operator norm).
- Assume the MoLEx adapters are small: $\|\Delta W'_k\| \leq \xi$ with $\xi > 0, \forall\, k = \overline{1, L}$. Assume the matrix A above is invertible. Assume bounded intermediate states $\mathbb{E}[\|x_t\|^2] \leq B < \infty$ and *pretrained mixing closeness*:

  $$\mathbb{E}\left[\left\|f_{h_t}(W_{h_t}x_t) - f_t(W_t x_t)\right\|^2\right] \leq \gamma^2 \mathbb{E}[\|x_t\|^2] + \beta^2$$

  for small $\gamma, \beta \geq 0$ independent of ξ.

These conditions are satisfied by common activation functions (ReLU, GeLU, SwiGLU, etc.) on bounded domains or under mild regularization. Furthermore, the final assumption is well-motivated: recent work [63] shows that the outputs of Transformer blocks exhibit high similarity, which has driven further investigation into model pruning. Motivated by this observation, we impose a bound on the block outputs, referred to as the "*pretrained mixing closeness*" assumption.

We construct the state-based adapters $\{\Delta W_k\}_{k=1}^{L}$ exactly as in the linear case: solve the invertible linear system derived above (possible since $\alpha > \frac{1}{2}$) to match the *first-order* effective updates assuming identity Jacobians.

Consider a single update step t. By the Taylor theorem with integral remainder, for any layer k,

$$\begin{aligned} f_k((W_k + \Delta W_k')z_t) &= f_k(W_k z_t) + \nabla f_k(W_k z_t)\,(\Delta W_k' z_t) \\ &\quad + \int_0^1 (1-s)\nabla^2 f_k(W_k z_t + s\Delta W_k' z_t)\,ds\,(\Delta W_k' z_t \otimes \Delta W_k' z_t) \\ &= f_k(W_k z_t) + \nabla f_k(W_k z_t)\,(\Delta W_k' z_t) + R_k(z_t), \end{aligned} \tag{15}$$

where $\otimes$ denotes the appropriate bilinear action of the Hessian.

The remainder $R_k(z_t)$ satisfies

$$\|R_k(z_t)\| \le \frac{H}{2}\|\Delta W_k' z_t\|^2 \le \frac{H}{2}\xi^2\|z_t\|^2.$$

Substituting into the MoLEx update, we obtain

$$\begin{aligned} z_{t+1} = z_t &+ \alpha\Big[f_t(W_t z_t) + \nabla f_t(W_t z_t)(\Delta W_t' z_t) + R_t(z_t)\Big] \\ &+ (1-\alpha)\Big[f_{h_t}(W_{h_t} z_t) + \nabla f_{h_t}(W_{h_t} z_t)(\Delta W_{h_t}' z_t) + R_{h_t}(z_t)\Big]. \end{aligned} \tag{16}$$

Recall that the generalize state-based update is presented by

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha\,\Delta W_t x_t + (1-\alpha)\,\Delta W_{h_t} x_t.$$

Define the *linearized MoLEx control* at step t as

$$\tilde{M}_t := \alpha\,\nabla f_t(W_t z_t)\,\Delta W_t' + (1-\alpha)\,\nabla f_{h_t}(W_{h_t} z_t)\,\Delta W_{h_t}'.$$

The MoLEx update expands to

$$\begin{aligned} z_{t+1} &= z_t + \alpha f_t(W_t z_t) + (1-\alpha) f_{h_t}(W_{h_t} z_t) + \tilde{M}_t z_t + \alpha R_t(z_t) + (1-\alpha) R_{h_t}(z_t) \\ &= z_t + f_t(W_t z_t) + (1-\alpha)[f_{h_t}(W_{h_t} z_t) - f_t(W_t z_t)] + \tilde{M}_t z_t + \alpha R_t(z_t) + (1-\alpha) R_{h_t}(z_t). \end{aligned} \tag{17}$$

Assume that in step t, we have $z_t = x_t$, then the difference $z_{t+1} - x_{t+1}$ consists of three terms:

1. Base mixing error: $(1-\alpha)[f_{h_t}(W_{h_t} x_t) - f_t(W_t x_t)]$, with L^2 norm $\le |1-\alpha|^2(\gamma^2 B + \beta^2)$.
2. Control mismatch: $\tilde{M}_t x_t - (\alpha\Delta W_t + (1-\alpha)\Delta W_{h_t})x_t$. By linear-case construction and $\|\nabla f_k\| \le L_f$, this is $O(\xi\sqrt{B})$.
3. Remainder: $\|\alpha R_t + (1-\alpha) R_{h_t}\| = O(\xi^2 B)$.

Applying the Cauchy-Schwarz inequality $\|\sum_{i=1}^3 a_i\|^2 \le 3\sum\|a_i\|^2$ over the three terms, taking expectations, and absorbing higher-order terms yields the per-step bound. Then, for each t, we have

$$\mathbb{E}\left[\|z_{t+1} - x_{t+1}\|^2\right] \le C\Big(|1-\alpha|^2(\gamma^2 + \beta^2/B) + \xi^2\Big),$$

with constant $C > 0$ depending on α, ξ, L_f, H, B. As $\xi \to 0$, and $\gamma, \beta \to 0$ (or $|1-\alpha| \to 0$), the trajectories converge uniformly in L^2:

$$\max_t \mathbb{E}\left[\|z_t - x_t\|^2\right] \to 0.$$

Accordingly, a generalized state-based cross-layer communication mechanism suffices to approximate the MoLEx representation while achieving a markedly reduced computational cost. This completes the proof. □

A.2 Proposition 4.2

Proof. We analyze the forward dynamics by bounding the growth of hidden state norms across layers. The key idea is to show that MoC's adaptive routing mechanism can select control modules with smaller operator norms, leading to improved stability. For Parallel Control with $\alpha = 1$ (no routing), the update is:

$$x_{t+1}^c = x_t^c + f_t(W_t x_t^c) + \Delta W_t x_t^c. \tag{18}$$

Taking norms and applying the triangle inequality:

$$\begin{aligned} \|x_{t+1}^c\| &\le \|x_t^c\| + \|f_t(W_t x_t^c)\| + \|\Delta W_t x_t^c\| \\ &\le \|x_t^c\| + L_f\|x_t^c\| + \|\Delta W_t\|\|x_t^c\|. \end{aligned} \tag{19}$$

where we used the assumption that $\|f_t(W_t x_t^c)\| \le L_f\|x_t^c\|$ (bounded nonlinearity with Lipschitz constant L_f).

Using the low-rank factorization $\Delta W_t = B_t A_t$ with $\|B_t\| \le \beta_t$ and $\|A_t\| \le \gamma$:

$$\|x_{t+1}^c\| \le (1 + L_f + \beta_t\gamma)\|x_t^c\|. \tag{20}$$

By taking the expectation over data distribution, we have:

$$\mathbb{E}[\|x_{t+1}^c\|/\|x_t^c\|] \le (1 + L_f + \beta_t\gamma) \tag{21}$$

Similarly, for the MoC dynamic with mixing parameter $\alpha \in [0, 1]$ and TopK routing, we imply:

$$x_{t+1} = x_t + f_t(W_t x_t) + \alpha\Delta W_t x_t + (1-\alpha)\sum_{k\in S_K(x_t)} \pi_k(x_t)\Delta W_k x_t \tag{22}$$

where $S_K(x_t) \subseteq \{1, \dots, L\}$ are the top-K selected experts and $\sum_{k\in S_K} \pi_k(x_t) = 1$. Similarly to Parallel Control, we take norms for MoC:

$$\begin{aligned} \|x_{t+1}\| &\le \|x_t\| + \|f_t(W_t x_t)\| + \alpha\|\Delta W_t x_t\| + (1-\alpha)\left\|\sum_{k\in S_K} \pi_k(x_t)\Delta W_k x_t\right\| \\ &\le \|x_t\| + L_f\|x_t\| + \alpha\beta_t\gamma\|x_t\| + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\|\Delta W_k x_t\| \end{aligned} \tag{23}$$

where the last step uses convexity of the norm: $\|\sum_k \pi_k v_k\| \le \sum_k \pi_k\|v_k\|$ for $\pi_k \ge 0, \sum_k \pi_k = 1$. Using $\|\Delta W_k\| \le \beta_k\gamma$, we imply:

$$\begin{aligned} \|x_{t+1}\| &\le \|x_t\| + L_f\|x_t\| + \alpha\beta_t\gamma\|x_t\| + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\beta_k\gamma\|x_t\| \\ &= (1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma)\,\|x_t\| \end{aligned} \tag{24}$$

where we define the effective routing norm:

$$\beta_K(x_t) := \sum_{k\in S_K(x_t)} \pi_k(x_t)\beta_k. \tag{25}$$

Taking expectations over the data distribution $\mathcal{D}$:

$$\mathbb{E}[\|x_{t+1}\|/\|x_t\|] \le (1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\mathbb{E}[\beta_K(x_t)|x_t]\gamma) \tag{26}$$

By routing learns to select experts with smaller operator norms on average:

$$\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t, \tag{27}$$

then the norm bound forward pass will be smaller than Parallel Control:

$(1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\mathbb{E}[\beta_K(x_t)|x_t]\gamma) < (1 + L_f + \alpha\beta_t\gamma + (1-\alpha)\beta_t\gamma) = (1 + L_f + \beta_t\gamma)$, which imply that MoC achieves improved forward stability compared to conventional control when routing is effective. □

Noting that with the assumption $\mathbb{E}[\beta_K(x_t)|x_t] < \beta_t$, we employ the load-balancing loss [36, 37] to encourage balanced token routing across experts. This promotes more effective utilization of high-capacity experts, thereby increasing the likelihood of selecting suitable experts that meet the assumption.

A.3 Corollary 4.3

Proof. We analyze how control affects gradient flow during backpropagation through the network. The backward gradient through block t is computed via the chain rule:

$$\frac{\partial \mathcal{L}}{\partial x_t} = J_t^\top \frac{\partial \mathcal{L}}{\partial x_{t+1}}, \tag{28}$$

then for MoC, differentiating the update equation gives:

$$\begin{aligned} J_t &= \frac{\partial}{\partial x_t}\left[x_t + f_t(W_t x_t) + \alpha \Delta W_t x_t + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\Delta W_k x_t\right] \\ &= I + \frac{\partial f_t(W_t x_t)}{\partial x_t} + \alpha \Delta W_t + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\Delta W_k \\ &\quad + (1-\alpha)\sum_{k\in S_K} \frac{\partial \pi_k(x_t)}{\partial x_t} \otimes (\Delta W_k x_t) \end{aligned} \tag{29}$$

For simplicity, we focus on the dominant terms and assume routing weights are approximately constant locally (i.e., the gradient w.r.t. π_k is small), giving:

$$J_t \approx I + \frac{\partial f_t(W_t x_t)}{\partial x_t} + \alpha \Delta W_t + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\Delta W_k \tag{30}$$

Then we take norms and using the Lipschitz property of f_t with constant L_f:

$$\begin{aligned} \|J_t\| &\le 1 + \left\|\frac{\partial f_t(W_t x_t)}{\partial x_t}\right\| + \alpha\|\Delta W_t\| + (1-\alpha)\left\|\sum_{k\in S_K} \pi_k(x_t)\Delta W_k\right\| \\ &\le 1 + L_f\|W_t\| + \alpha\|\Delta W_t\| + (1-\alpha)\sum_{k\in S_K} \pi_k(x_t)\|\Delta W_k\| \\ &\le 1 + L_f\|W_t\| + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma \end{aligned} \tag{31}$$

Under the assumptions of Proposition 4.2, we obtain:

$$1 + L_f\|W_t\| + \alpha\beta_t\gamma + (1-\alpha)\beta_K(x_t)\gamma < 1 + L_f\|W_t\| + \beta_t\gamma,$$

which indicates that the Jacobian norm of MoC is more tightly bounded than that of Parallel Control, leading to a more stable gradient flow. □

In addition to the stability analyses of the representation and its gradient dynamics, we assess the fidelity of the learned representations. Assume $\Delta W^* x$ denote the optimal control signal that yields task-optimal representations. Given a MoC signal $g_{\text{moc}}(x)$, then the expected control-approximation error over data distribution is defined as follows:

$$\bar{\epsilon}^2 \coloneqq \mathbb{E}\left[\|g_{\text{moc}}(x) - \Delta W^* x\|^2\right]. \tag{32}$$

A.4 Proposition 4.4

Proof. Given input x, then control error of Parallel Control is:

$$\delta_t^c(x) = \Delta W_t x - \Delta W^* x, \tag{33}$$

then control update at block t of MoC can be written as follows:

$$g_{\text{moc}}(x) = \alpha \Delta W_t x + (1-\alpha)\sum_{k\in S_K(x)} \pi_k(x)\Delta W_k x. \tag{34}$$

Next, the control representation error of MoC is:

$$\begin{aligned}\delta_t(x) &= g_{\text{moc}}(x) - \Delta W^* x \\ &= \alpha[\Delta W_t x - \Delta W^* x] + (1-\alpha)\sum_{k\in S_K(x)} \pi_k(x)[\Delta W_k x - \Delta W^* x] \quad (35)\\ &= \alpha\delta_t^c(x) + (1-\alpha)\sum_{k\in S_K(x)} \pi_k(x)\delta_k^c(x), \quad (36)\end{aligned}$$

where we have $\sum_{k\in S_K(x)} \pi_k(x) = 1, \pi_k(x) \geq 0, \forall\, k = \overline{1, L}$. Similarly,

By convexity of $\|\cdot\|^2$ (Jensen's inequality):

$$\begin{aligned}\|\delta_t(x)\|^2 &= \left\|\alpha\delta_t^c(x) + (1-\alpha)\sum_{k\in S_K(x)} \pi_k(x)\delta_k^c(x)\right\|^2 \\ &\leq \alpha\|\delta_t^c(x)\|^2 + (1-\alpha)\left\|\sum_{k\in S_K(x)} \pi_k(x)\delta_k^c(x)\right\|^2 \quad (37)\\ & \quad (38)\end{aligned}$$

Taking expectation over $x \sim \mathcal{D}$, $t = \overline{1, L}$:

$$\begin{aligned}\bar{\epsilon}^2 &= \mathbb{E}[\|\delta_t(x)\|^2] \\ &\leq \alpha\mathbb{E}[\|\delta_t^c(x)\|^2] + (1-\alpha)\mathbb{E}\left[\left\|\sum_{k\in S_K(x)} \pi_k(x)\delta_k^c(x)\right\|^2\right] \\ &= \alpha\bar{\epsilon}_c^2 + (1-\alpha)\bar{\epsilon}_K^2 \quad (39)\end{aligned}$$

This completes the proof. □

B Experimental Details

This section provides comprehensive details about our experimental setup, including hyperparameter configurations, dataset specifications, and implementation details for all models evaluated in this work.

B.1 Dataset Details

To comprehensively evaluate the effectiveness of our proposed method, we conduct experiments across multiple benchmark datasets covering different NLP task categories.

Natural Language Understanding (NLU) For NLU tasks, we employ the GLUE benchmark [49], a well-established benchmark suite that includes datasets such as SST-2, MRPC, CoLA, QNLI, RTE, and STS-B. These tasks collectively evaluate model performance on grammaticality judgment, sentiment classification, semantic similarity, and textual entailment across domain distributions.

Natural Language Generation and Commonsense Reasoning (NLG) For evaluating NLG and commonsense reasoning capabilities, we utilize eight benchmark datasets: BoolQ [42], PIQA [43], Social IQa [44], HellaSwag [45], WinoGrande [46], ARC-Easy/ Challenge [47], and OpenBookQA [48]. These datasets assess model understanding of complex reasoning, physical commonsense, and social dynamics.

B.2 Model Architecture and Training Configuration

Encoder-Only Models. We utilized some Encoder-only models as the model backbones, particularly RoBERTa [64] and DeBERTa [65], on different model size (base and large), incorporating LoRA, DoRA, MoLEx and Parallel Control methods as the baselines. The detailed hyperparameter configurations for BERT-based models are illustrated in Table 8.

Decoder-Only Models. We utilized some Large scale models as the backbones, including LLaMA2-7B [51], LLaMA3-8B [52], Mistral-7B Instruct [53], Qwen2.5-14B Instruct [54]. These pretrained models are also finetuned across our methods as well as the baselines. The detailed hyperparameter configurations for the decoder-only models are illustrated in Table 9. For routing, we consider two variants: (1) a linear router and (2) a cosine router with minimal extra parameters via a low-dimensional projection, which expands the latent space for richer representations.

Vision Tasks. We finetune ViT-Tiny/Base backbones on VTAB datasets, under batch size 128, 50 epochs, learning rate 0.005 to ensure fair comparison.

Weight-based Configurations. Across weight-based methods, we apply rank-32 LoRA modules to W_Q, W_K and W_V in attention and $W_{\text{up}}, W_{\text{down}}$ in the FFN. For MoLEx, as it additionally introduces a lightweight, learnable, routing gate, we use Top-K=1, the best-performing configuration reported for MoLEx, and keep the remaining training settings identical across methods and modalities as shown in Table 8 and Table 9. Because MoLEx may exceeds the memory limit, in this case, we distribute it across two GPUs H100 80GB rather than reducing its batch size or rank for fair comparison.

MoC Configurations. For efficient Transformer training, we adopt batch-level routing in the MoC framework. Instead of routing each token independently, which would result in token-specific control assignments and hinder efficient batched computation, we make a unified routing decision for all tokens in a batch. To obtain the batch-level TopK controls, we aggregate token-level gate outputs using either "mode" or "mean" aggregation. The mode strategy selects the most frequently activated controls across tokens, while the mean strategy averages token-level probability vectors and chooses the controls with the highest averaged probabilities. In our experiments, we use either mode- or mean-based batch routing. The load-balancing weight is selected from $[0.001, 0.02]$, and the mixing coefficient is set to $\alpha \in \{0.6, 0.7, \ldots, 0.9, 0.95\}$, following Theorem 4.1.

Table 8: Hyperparameter configurations for DeBERTa and RoBERTa models on GLUE tasks. Common settings include AdamW optimizer, linear learning rate schedule, warmup ratio of 0.06, and sequence length of 128 tokens.

Model	Dataset	SST-2	MRPC	CoLA	QNLI	RTE	STS-B
	Optimizer	AdamW					
	WarmUp Ratio	0.06					
	LR Schedule	Linear					
	Max Seq Length	128					
$\text{DeBERTa}_{\text{base}}$	Batch Size	128	128	64	256	128	128
	Epochs	50	30	80	25	80	40
	Learning Rate	5e-4	4e-4	3e-4	4e-4	4e-4	4e-4
$\text{DeBERTa}_{\text{large}}$	Batch Size	64	32	32	32	64	32
	Epochs	20	30	20	10	20	10
	Learning Rate	4e-4	3e-4	5e-4	3e-4	4e-4	3e-4
$\text{RoBERTa}_{\text{base}}$	Batch Size	128	128	64	256	128	128
	Epochs	50	30	80	25	80	40
	Learning Rate	5e-4	4e-4	3e-4	4e-4	4e-4	4e-4
$\text{RoBERTa}_{\text{large}}$	Batch Size	64	32	32	32	64	32
	Epochs	20	30	20	10	20	10
	Learning Rate	4e-4	3e-4	5e-4	3e-4	4e-4	3e-4

Compute Resources The compute resource used in our experiments consists of 8 NVIDIA H100 GPUs, each with 80GB of memory.

B.3 Additional Experiments

Efficiency Comparison. As shown in Table 1, we compare MoC with recent strong baselines selected primarily for competitive performance rather than identical parameter budgets; thus, these results do not fully characterize parameter efficiency. To provide a fairer comparison, Table 12

Table 9: Hyperparameter settings and Mixture-of-Control configuration for LLaMA2-7B, LLaMA3-8B, Mistralv3-7B, and Qwen2.5-14B. Settings are chosen to balance performance and training efficiency.

Parameter	**LLaMA2-7B**	**LLaMA3-8B**	**Mistralv3-7B**	**Qwen2.5-14B**
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Batch size	16	16	16	12
Gradient accumulation steps	2	2	2	2
Max sequence length	256	256	256	256
Projection dimension	256	256	256	256
Training epochs	3	3	3	3
Weight decay	0.01	0.01	0.01	0.01

Table 10: GLUE results with DeBERTa-Base/Large backbones. Values are medians of five runs (different seeds); * marks previously published numbers. Best results are in **bold**.

Model	**Method**	**SST-2**	**MRPC**	**CoLA**	**QNLI**	**RTE**	**STS-B**	**Avg**
DeBERTa-Base	LoRA*	96.2	89.4	70.4	94.3	83.4	91.0	87.5
	DoRA*	96.0$_{\downarrow 0.2}$	90.2$_{\uparrow 0.8}$	71.2$_{\uparrow 0.8}$	94.3$_{\uparrow 0.0}$	84.0$_{\uparrow 0.6}$	91.0$_{\uparrow 0.0}$	87.8$_{\uparrow 0.3}$
	MoLEx	96.0$_{\downarrow 0.2}$	90.1$_{\uparrow 0.7}$	71.0$_{\uparrow 0.6}$	94.3$_{\uparrow 0.0}$	85.8$_{\uparrow 2.4}$	91.4$_{\uparrow 0.4}$	88.1$_{\uparrow 0.6}$
	Control	96.3$_{\uparrow 0.1}$	89.8$_{\uparrow 0.4}$	71.3$_{\uparrow 0.9}$	94.4$_{\uparrow 0.1}$	84.8$_{\uparrow 1.4}$	91.5$_{\uparrow 0.5}$	88.0$_{\uparrow 0.5}$
	MoC	**96.9**$_{\uparrow 0.7}$	**91.0**$_{\uparrow 1.6}$	**71.8**$_{\uparrow 1.4}$	**94.5**$_{\uparrow 0.2}$	**87.8**$_{\uparrow 4.4}$	**91.9**$_{\uparrow 0.9}$	**89.0**$_{\uparrow 1.5}$
DeBERTa-Large	LoRA*	96.9	91.8	75.0	95.9	91.0	92.7	90.5
	DoRA*	96.6$_{\downarrow 0.3}$	92.1$_{\uparrow 0.3}$	76.9$_{\uparrow 1.9}$	95.8$_{\downarrow 0.1}$	91.1$_{\uparrow 0.1}$	92.6$_{\downarrow 0.1}$	90.9$_{\uparrow 0.4}$
	MoLEx	96.6$_{\downarrow 0.3}$	91.9$_{\uparrow 0.1}$	75.9$_{\uparrow 0.9}$	95.9$_{\uparrow 0.0}$	91.6$_{\uparrow 0.6}$	92.7$_{\uparrow 0.0}$	90.8$_{\uparrow 0.3}$
	Control	96.7$_{\downarrow 0.2}$	**92.7**$_{\uparrow 0.9}$	76.4$_{\uparrow 1.4}$	95.9$_{\uparrow 0.0}$	90.0$_{\downarrow 1.0}$	92.3$_{\downarrow 0.4}$	90.6$_{\uparrow 0.1}$
	MoC	**97.3**$_{\uparrow 0.4}$	**92.7**$_{\uparrow 0.9}$	**77.0**$_{\uparrow 2.0}$	**96.2**$_{\uparrow 0.3}$	**93.9**$_{\uparrow 2.9}$	**92.9**$_{\uparrow 0.2}$	**91.7**$_{\uparrow 1.2}$

evaluates representative methods under matched parameter budgets on Mistralv3-7B-Instruct and Qwen2.5-14B-Instruct. Under this setting, MoC maintains memory usage close to LoRA while improving average accuracy by **8.0%** and **3.2%**, respectively. Compared with Control, MoC introduces only modest routing overhead while enabling cross-layer communication. It also avoids the additional pretrained-block forward passes required by full-block communication methods such as MoLEx. On Qwen2.5-14B-Instruct, MoC uses nearly half the training memory of MoLEx while achieving **2.5%** higher average accuracy.

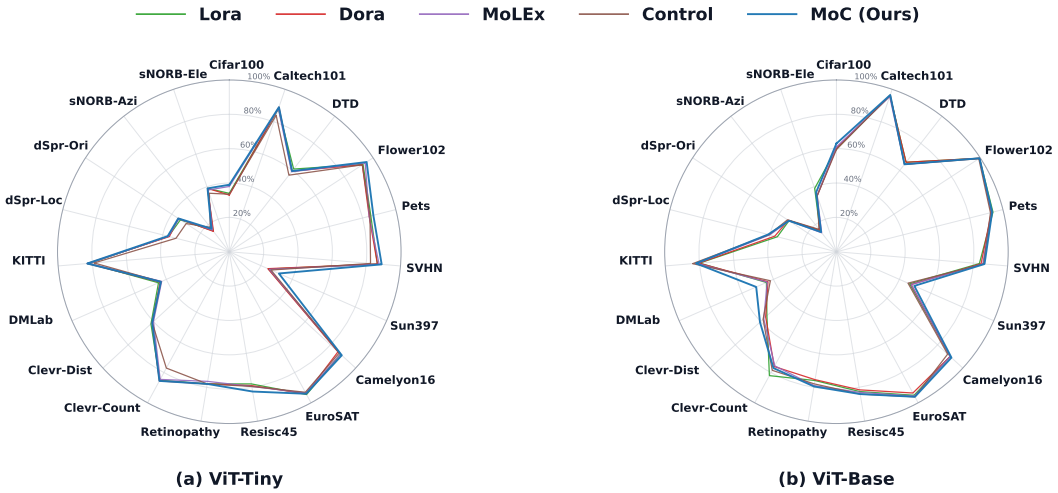
Batch Routing Scheme. As discussed in MoC Configurations, we aggregate token-level gate outputs using either "mode" or "mean" aggregation. Here we further provide a comparison of batch-level routing schemes in Table 13. Mean aggregation improves average accuracy over mode aggregation by **0.3%** on Mistralv3-7B-Instruct and **1.2%** on Qwen2.5-14B-Instruct, indicating moderate model-dependent sensitivity to the routing scheme.

Inference Throughput. We measure the inference throughput of different LLMs across batch sizes, as shown in Figure 2. Given 100 samples, we run batched inference and compute throughput as the number of generated tokens divided by the inference time for each batch. We then report the average throughput across all batches for each model.

Routing Behavior. To better understand the routing behavior of MoC, we visualize the selected experts across layers for both Attention and FFN blocks under different routing budgets. The routing patterns show that, at a given timestep, MoC is not limited to the local control associated with the current layer. Instead, it often selects experts from non-local layers, suggesting that the model learns to dynamically aggregate cross-layer information for richer intermediate representations. We report the visualizations for LLaMA3-8B under Top-1, Top-3, and Top-7 routing, with similar routing patterns observed across other backbone models (see Figures 5 to 13).

Table 11: VTAB-1K results evaluated on Top-1 accuracy. All methods are applied to finetune the same set of LoRA parameters on ViT-Tiny/Base pre-trained with ImageNet-21K dataset.

	Natural							Specialized				Structured								
Method	Cifar100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon16	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI	dSpr-Loc	dSpr-Ori	sNORB-Azi	sNORB-Ele	**Mean**
ViT-Tiny																				
LoRA	34.0	88.0	**61.0**	93.0	85.0	86.5	25.0	88.0	**94.5**	78.0	**78.0**	85.0	**62.0**	**45.0**	82.0	36.5	34.0	15.5	**39.0**	63.7
DoRA	33.0$_{\downarrow 1.0}$	**89.0**$_{\uparrow 1.0}$	59.5$_{\downarrow 1.5}$	92.5$_{\downarrow 0.5}$	84.0$_{\downarrow 1.0}$	86.5$_{\uparrow 0.0}$	24.5$_{\downarrow 0.5}$	86.0$_{\downarrow 2.0}$	94.0$_{\downarrow 0.5}$	79.0$_{\uparrow 1.0}$	**78.0**$_{\uparrow 0.0}$	**85.5**$_{\uparrow 0.5}$	61.0$_{\downarrow 1.0}$	44.0$_{\downarrow 1.0}$	**83.0**$_{\uparrow 1.0}$	36.0$_{\downarrow 0.5}$	**35.5**$_{\uparrow 1.5}$	15.0$_{\downarrow 0.5}$	**39.0**$_{\uparrow 0.0}$	63.4$_{\downarrow 0.3}$
MoLEx	38.0$_{\uparrow 4.0}$	88.0$_{\uparrow 0.0}$	59.0$_{\downarrow 2.0}$	94.5$_{\uparrow 1.5}$	84.0$_{\downarrow 1.0}$	87.5$_{\uparrow 1.0}$	25.5$_{\uparrow 0.5}$	88.0$_{\uparrow 0.0}$	93.5$_{\downarrow 1.0}$	79.5$_{\uparrow 1.5}$	76.5$_{\downarrow 1.5}$	84.5$_{\downarrow 0.5}$	60.0$_{\downarrow 2.0}$	43.0$_{\downarrow 2.0}$	82.0$_{\uparrow 0.0}$	36.5$_{\uparrow 0.0}$	35.0$_{\uparrow 1.0}$	16.0$_{\uparrow 0.5}$	38.0$_{\downarrow 1.0}$	63.6$_{\downarrow 0.1}$
Control	33.5$_{\downarrow 0.5}$	84.0$_{\downarrow 4.0}$	56.5$_{\downarrow 4.5}$	92.5$_{\downarrow 0.5}$	84.5$_{\downarrow 0.5}$	82.5$_{\downarrow 4.0}$	26.5$_{\uparrow 1.5}$	87.0$_{\downarrow 1.0}$	93.0$_{\downarrow 1.5}$	79.5$_{\uparrow 1.5}$	**78.0**$_{\uparrow 0.0}$	77.0$_{\downarrow 8.0}$	61.0$_{\downarrow 1.0}$	43.5$_{\downarrow 1.5}$	79.0$_{\downarrow 3.0}$	32.0$_{\downarrow 4.5}$	30.0$_{\downarrow 4.0}$	**18.0**$_{\uparrow 2.5}$	36.0$_{\downarrow 3.0}$	61.8$_{\downarrow 1.9}$
MoC	**39.0**$_{\uparrow 5.0}$	**89.0**$_{\uparrow 1.0}$	59.5$_{\downarrow 1.5}$	**95.5**$_{\uparrow 2.5}$	**86.5**$_{\uparrow 1.5}$	**89.0**$_{\uparrow 2.5}$	**31.5**$_{\uparrow 6.5}$	**89.0**$_{\uparrow 1.0}$	94.0$_{\downarrow 0.5}$	**82.5**$_{\uparrow 4.5}$	**78.0**$_{\uparrow 0.0}$	**85.5**$_{\uparrow 0.5}$	61.0$_{\downarrow 1.0}$	43.5$_{\downarrow 1.5}$	**83.0**$_{\uparrow 1.0}$	**37.0**$_{\uparrow 0.5}$	**35.5**$_{\uparrow 1.5}$	17.0$_{\uparrow 1.5}$	**39.0**$_{\uparrow 0.0}$	**65.0**$_{\uparrow 1.3}$
ViT-Base																				
LoRA	60.0	96.0	**66.0**	**99.5**	**94.0**	83.5	46.5	90.0	95.0	82.5	76.0	**82.0**	55.0	44.5	80.0	35.5	33.0	15.5	**39.0**	67.0
DoRA	60.5$_{\uparrow 0.5}$	96.0$_{\uparrow 0.0}$	**66.0**$_{\uparrow 0.0}$	**99.5**$_{\uparrow 0.0}$	93.0$_{\downarrow 1.0}$	84.5$_{\uparrow 1.0}$	47.5$_{\uparrow 1.0}$	90.0$_{\uparrow 0.0}$	93.5$_{\downarrow 1.5}$	81.5$_{\downarrow 1.0}$	75.5$_{\downarrow 0.5}$	76.0$_{\downarrow 6.0}$	56.0$_{\uparrow 1.0}$	43.5$_{\downarrow 1.0}$	80.5$_{\uparrow 0.5}$	37.0$_{\uparrow 1.5}$	33.5$_{\uparrow 0.5}$	16.0$_{\uparrow 0.5}$	34.5$_{\downarrow 4.5}$	66.6$_{\downarrow 0.4}$
MoLEx	61.0$_{\uparrow 1.0}$	95.5$_{\downarrow 0.5}$	64.5$_{\downarrow 1.5}$	**99.5**$_{\uparrow 0.0}$	93.0$_{\downarrow 1.0}$	85.5$_{\uparrow 2.0}$	47.5$_{\uparrow 1.0}$	90.0$_{\uparrow 0.0}$	95.5$_{\uparrow 0.5}$	83.0$_{\uparrow 0.5}$	79.0$_{\uparrow 3.0}$	75.5$_{\downarrow 6.5}$	57.5$_{\uparrow 2.5}$	44.0$_{\downarrow 0.5}$	81.5$_{\uparrow 1.5}$	40.0$_{\uparrow 4.5}$	33.0$_{\uparrow 0.0}$	14.0$_{\downarrow 1.5}$	34.5$_{\downarrow 4.5}$	67.1$_{\uparrow 0.1}$
Control	59.5$_{\downarrow 0.5}$	**96.5**$_{\uparrow 0.5}$	64.5$_{\downarrow 1.5}$	99.0$_{\downarrow 0.5}$	92.5$_{\downarrow 1.5}$	86.0$_{\uparrow 2.5}$	45.5$_{\downarrow 1.0}$	88.0$_{\downarrow 2.0}$	**96.0**$_{\uparrow 1.0}$	**84.0**$_{\uparrow 1.5}$	78.0$_{\uparrow 2.0}$	78.5$_{\downarrow 3.5}$	58.0$_{\uparrow 3.0}$	42.0$_{\downarrow 2.5}$	**84.0**$_{\uparrow 4.0}$	40.5$_{\uparrow 5.0}$	**34.0**$_{\uparrow 1.0}$	**16.5**$_{\uparrow 1.0}$	34.5$_{\downarrow 4.5}$	67.2$_{\uparrow 0.2}$
MoC	**63.0**$_{\uparrow 3.0}$	**96.5**$_{\uparrow 0.5}$	64.5$_{\downarrow 1.5}$	**99.5**$_{\uparrow 0.0}$	93.5$_{\downarrow 0.5}$	**86.5**$_{\uparrow 3.0}$	**49.5**$_{\uparrow 3.0}$	**91.0**$_{\uparrow 1.0}$	**96.0**$_{\uparrow 1.0}$	**84.0**$_{\uparrow 1.5}$	**79.5**$_{\uparrow 3.5}$	77.0$_{\downarrow 5.0}$	**60.5**$_{\uparrow 5.5}$	**51.0**$_{\uparrow 6.5}$	82.0$_{\uparrow 2.0}$	**41.0**$_{\uparrow 5.5}$	33.0$_{\uparrow 0.0}$	14.5$_{\downarrow 1.0}$	36.5$_{\downarrow 2.5}$	**68.4**$_{\uparrow 1.4}$



(a) ViT-Tiny (b) ViT-Base

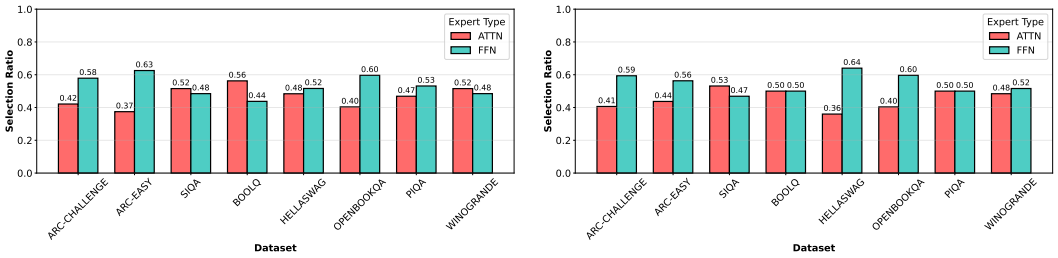
Figure 4: Benchmark comparison of LoRA, DoRA, MoLEx, Control, and MoC on VTAB across 19 datasets, shown separately for ViT-Tiny (left) and ViT-Base (right). Values indicate classification accuracy, with the outer ring corresponding to 100%.

Table 12: Results on commonsense reasoning benchmarks with Mistralv3-7B-Instruct and Qwen2.5-14B-Instruct backbones. Performance improvements and degradations are measured relative to the LoRA baseline for each backbone. Best task performance is shown in **bold**.

Model	Method	Trainable (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg	Time (h)	Memory (GB)
Mistralv3-7B	LoRA	0.49	67.8	83.0	77.8	89.4	79.3	78.7	64.9	79.2	77.5	5.8	54.3
	DoRA	0.50	68.1$_{\uparrow 0.3}$	82.4$_{\downarrow 0.6}$	77.7$_{\downarrow 0.1}$	88.1$_{\downarrow 1.3}$	78.7$_{\downarrow 0.6}$	77.1$_{\downarrow 1.6}$	64.3$_{\downarrow 0.6}$	79.4$_{\uparrow 0.2}$	77.0$_{\downarrow 0.5}$	9.5	64.7
	MoLEx	0.49	69.5$_{\uparrow 1.7}$	83.5$_{\uparrow 0.5}$	77.9$_{\uparrow 0.1}$	89.4$_{\uparrow 0.0}$	82.2$_{\uparrow 2.9}$	80.1$_{\uparrow 1.4}$	67.6$_{\uparrow 2.7}$	81.0$_{\uparrow 1.8}$	78.9$_{\uparrow 1.4}$	10.0	75.4
	Control	0.46	70.6$_{\uparrow 2.8}$	86.9$_{\uparrow 3.9}$	79.0$_{\uparrow 1.2}$	94.2$_{\uparrow 4.8}$	84.8$_{\uparrow 5.5}$	87.9$_{\uparrow 9.2}$	76.5$_{\uparrow 11.6}$	84.8$_{\uparrow 5.6}$	83.1$_{\uparrow 5.6}$	**4.0**	**53.2**
	MoC	0.48	**74.2**$_{\uparrow 6.4}$	**88.9**$_{\uparrow 5.9}$	**81.0**$_{\uparrow 3.2}$	**95.6**$_{\uparrow 6.2}$	**87.9**$_{\uparrow 8.6}$	**88.9**$_{\uparrow 10.2}$	**79.5**$_{\uparrow 14.6}$	**87.8**$_{\uparrow 8.6}$	**85.5**$_{\uparrow 8.0}$	6.0	55.6
Qwen2.5-14B	LoRA	0.43	72.9	88.0	80.7	94.4	87.3	92.8	85.8	93.6	86.9	7.9	60.9
	DoRA	0.44	73.8$_{\uparrow 0.9}$	88.8$_{\uparrow 0.8}$	81.5$_{\uparrow 0.8}$	95.6$_{\uparrow 1.2}$	88.2$_{\uparrow 0.9}$	94.3$_{\uparrow 1.5}$	86.6$_{\uparrow 0.8}$	89.0$_{\downarrow 4.6}$	87.2$_{\uparrow 0.3}$	14.0	90.4
	MoLEx	0.43	73.4$_{\uparrow 0.5}$	89.8$_{\uparrow 1.8}$	81.7$_{\uparrow 1.0}$	95.3$_{\uparrow 0.9}$	88.2$_{\uparrow 0.9}$	94.1$_{\uparrow 1.3}$	86.7$_{\uparrow 0.9}$	91.2$_{\downarrow 2.4}$	87.6$_{\uparrow 0.7}$	19.4	116.3
	Control	0.43	74.2$_{\uparrow 1.3}$	91.4$_{\uparrow 3.4}$	82.0$_{\uparrow 1.3}$	96.4$_{\uparrow 2.0}$	87.7$_{\uparrow 0.4}$	**96.7**$_{\uparrow 3.9}$	90.0$_{\uparrow 4.2}$	95.4$_{\uparrow 1.8}$	89.2$_{\uparrow 2.3}$	**6.5**	**58.7**
	MoC	0.43	**75.5**$_{\uparrow 2.6}$	**91.8**$_{\uparrow 3.8}$	**82.4**$_{\uparrow 1.7}$	**96.8**$_{\uparrow 2.4}$	**91.0**$_{\uparrow 3.7}$	96.6$_{\uparrow 3.8}$	**90.2**$_{\uparrow 4.4}$	**96.4**$_{\uparrow 2.8}$	**90.1**$_{\uparrow 3.2}$	8.0	61.3

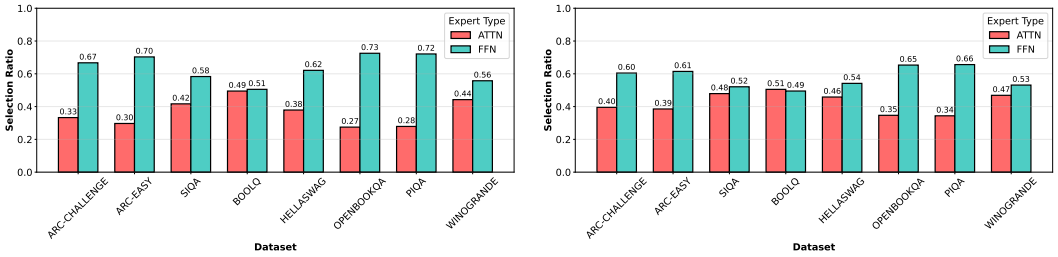
Table 13: Comparison of aggregation schemes for MoC on Mistralv3-7B-Instruct and Qwen2.5-14B-Instruct.

Model	Scheme	% Params	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg
Mistralv3-7B	Mode	0.48	73.6	87.5	80.3	95.5	86.6	**89.4**	**79.6**	**88.8**	85.2
	Mean	**0.48**	**74.2**	**88.9**	**81.0**	**95.6**	**87.9**	88.9	79.5	87.8	**85.5**
Qwen2.5-14B	Mode	0.43	73.8	89.8	81.9	96.5	89.2	95.5	88.7	95.4	88.9
	Mean	**0.43**	**75.5**	**91.8**	**82.4**	**96.8**	**91.0**	**96.6**	**90.2**	**96.4**	**90.1**



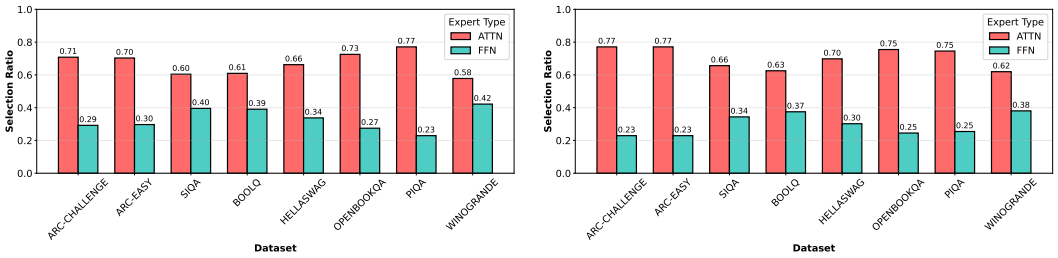
(a) Attention blocks (b) FFN blocks

Figure 5: Cross-type routing behavior in LLaMA3-8B under the Top-$K = 1$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.



(a) Attention blocks (b) FFN blocks

Figure 6: Cross-type routing behavior in LLaMA3-8B under the Top-$K = 3$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.



(a) Attention blocks (b) FFN blocks

Figure 7: Cross-type routing behavior in LLaMA3-8B under the Top-$K = 7$ setting, conditioned on the local control being from (a) Attention blocks or (b) FFN blocks.

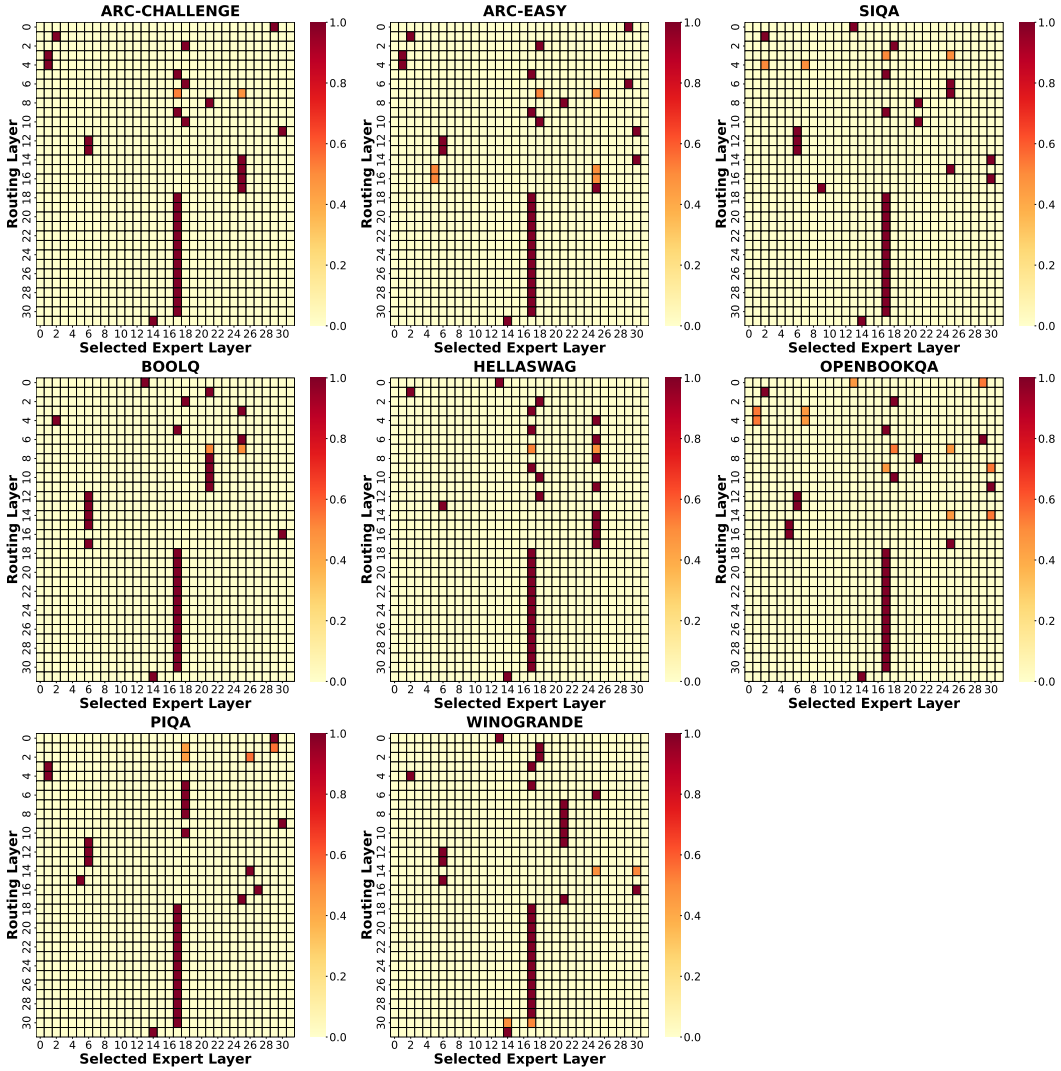


Figure 8: Routing behavior of MoC in LLaMA3-8B under Top-$K = 1$ routing when selecting Attention-block controls across layers.

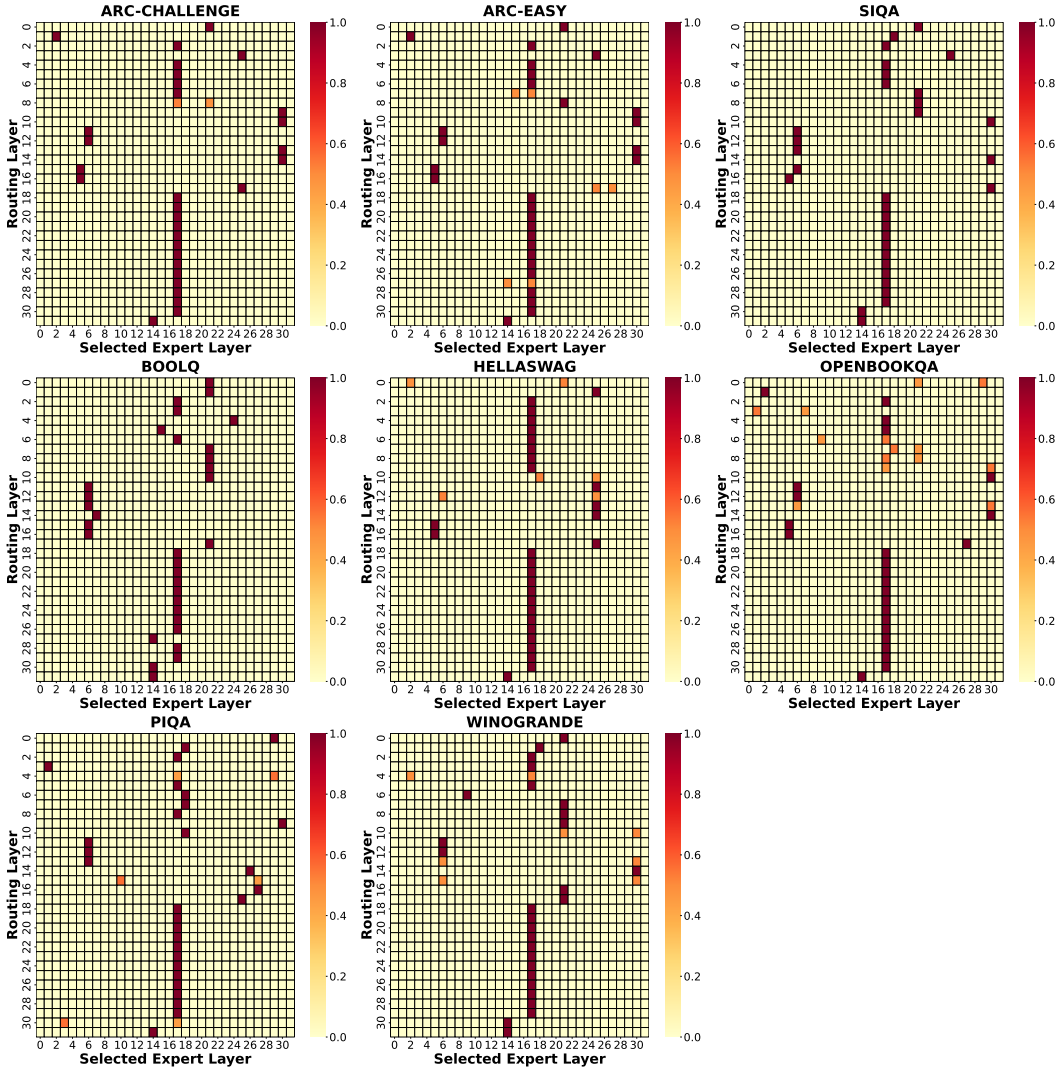


Figure 9: Routing behavior of MoC in LLaMA3-8B under Top-$K = 1$ routing when selecting FFN-block controls across layers.

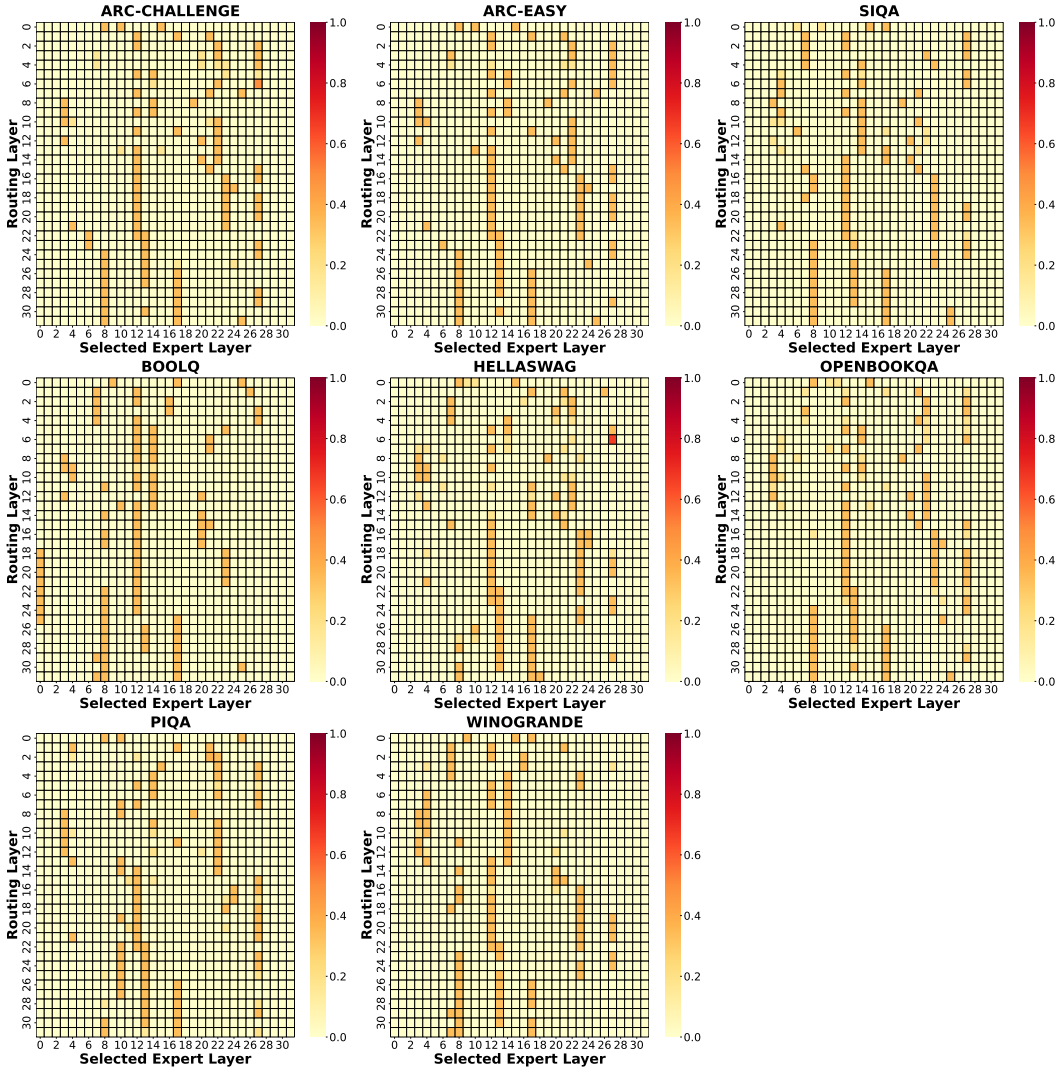


Figure 10: Routing behavior of MoC in LLaMA3-8B under Top-$K = 3$ routing when selecting Attention-block controls across layers.

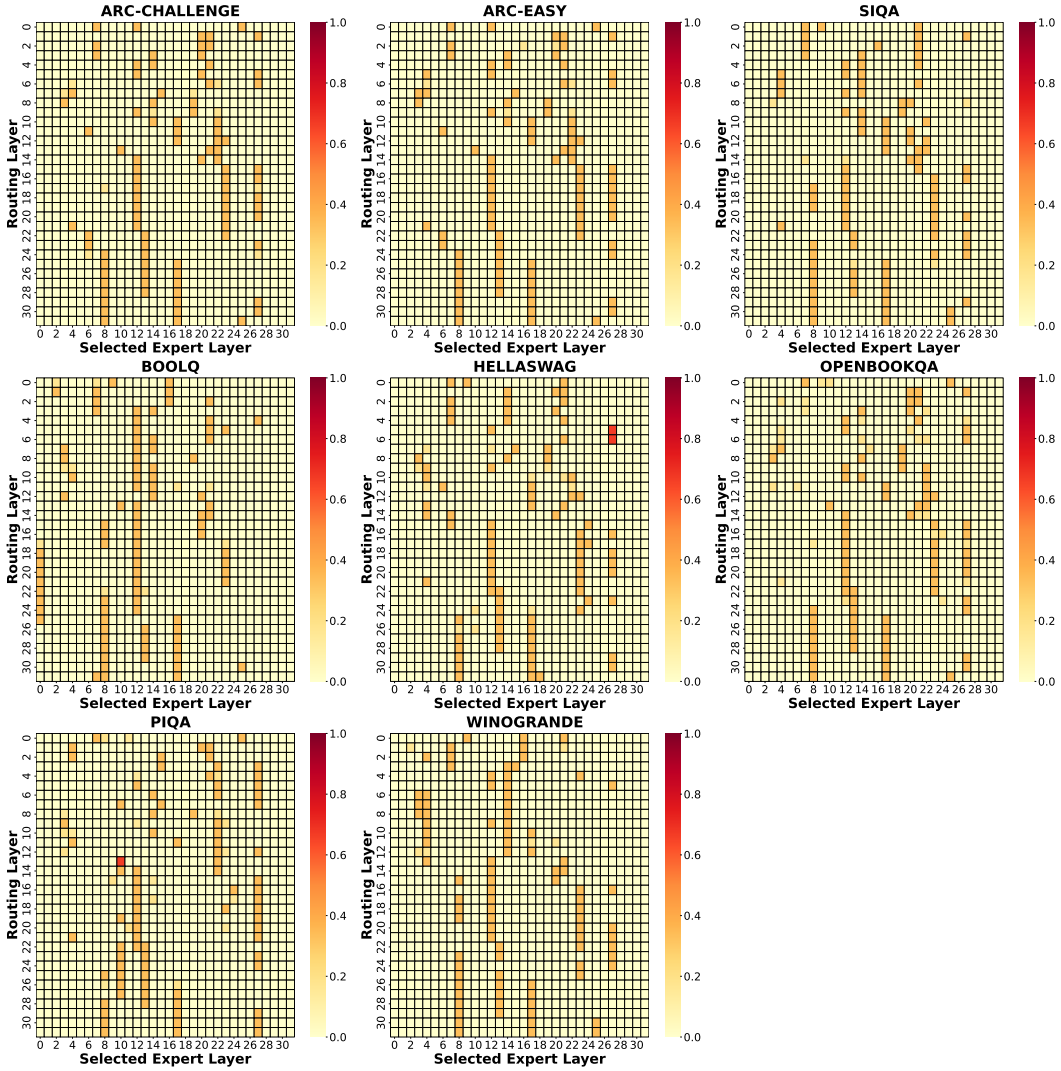


Figure 11: Routing behavior of MoC in LLaMA3-8B under Top-$K = 3$ routing when selecting FFN-block controls across layers.

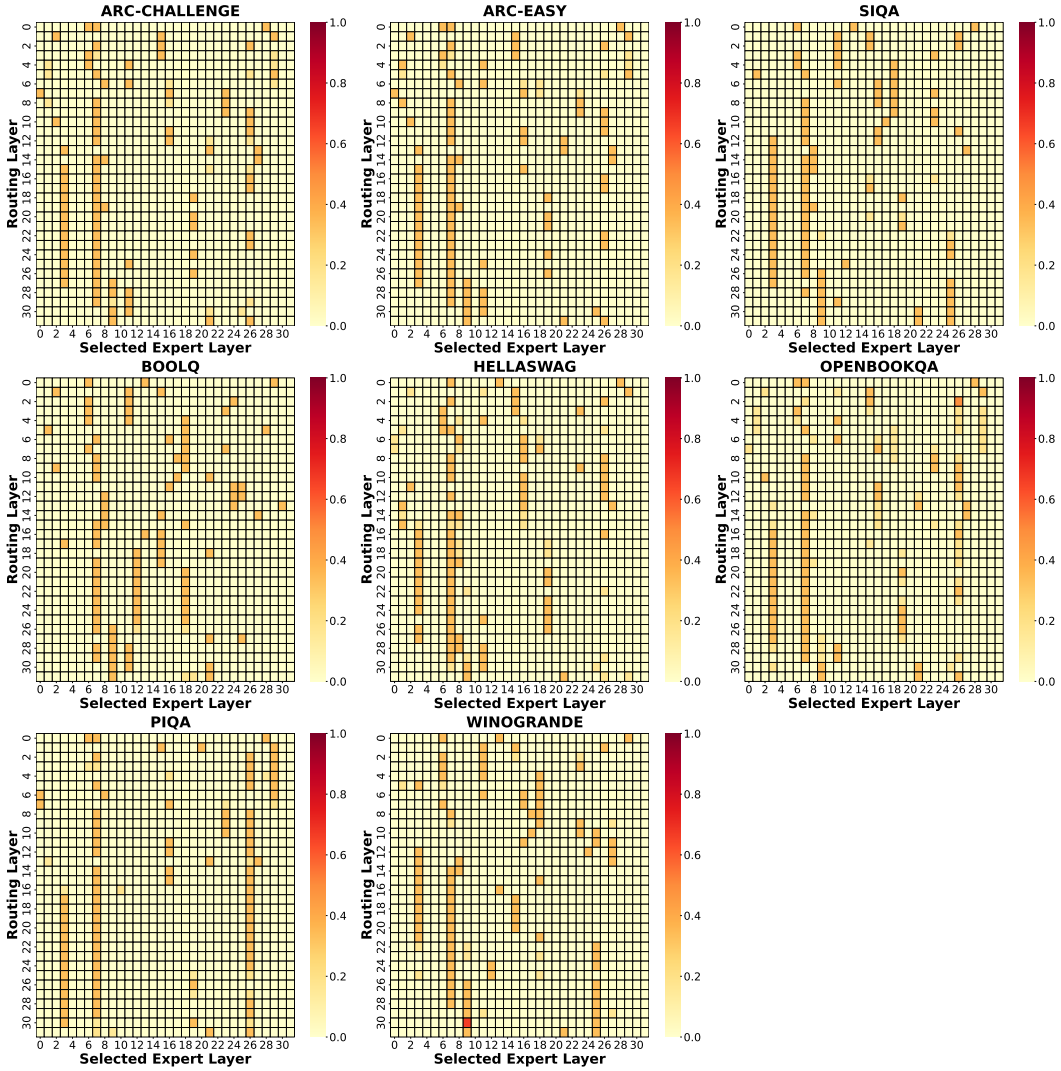


Figure 12: Routing behavior of MoC in LLaMA3-8B under Top-$K = 7$ routing when selecting Attention-block controls across layers.

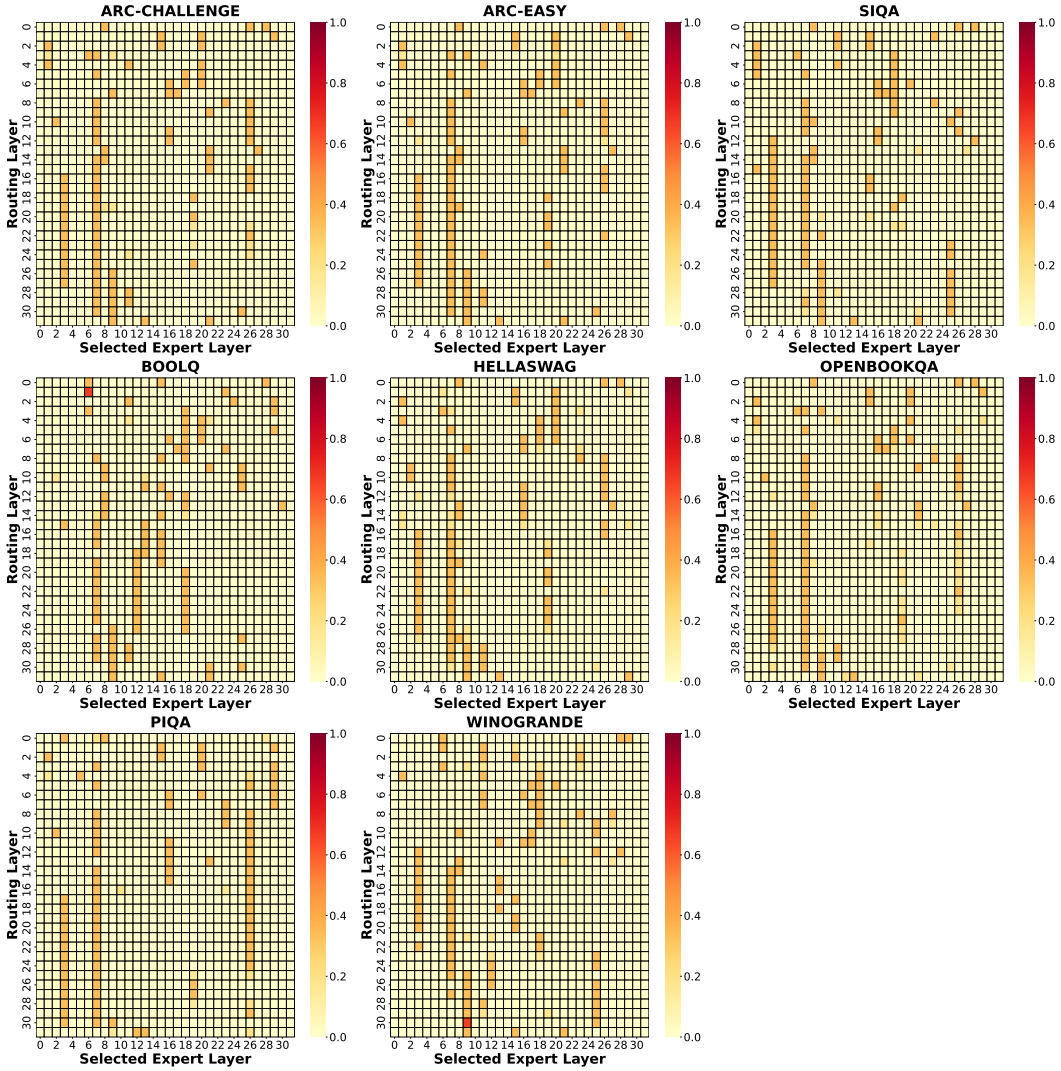


Figure 13: Routing behavior of MoC in LLaMA3-8B under Top-$K = 7$ routing when selecting FFN-block controls across layers.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are reflected accurately in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have already discussed the limitations of our work in Section 5 and in Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The full set of assumptions and proofs is given in Sections 4.1 and 4.3, as well as in Appendices.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the experimental details for reproducibility are specified in Section 5 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code and data release is subject to our company's internal approval process, which might not be completed before the camera-ready deadline. We plan to release them upon approval. Reproducibility details are provided in the Appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All the experimental details are specified in Section 5, and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Experimental results are tested multiple times to ensure stability and reliability (See Section 5).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided sufficient information on the computer resources in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and followed all the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Given the primarily theoretical nature of this work, we do not anticipate direct positive or negative societal impacts. The experiments are conducted solely to empirically support and validate the theoretical findings.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not release any data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The data and models used in the paper are properly credited. See Section 5 as well as Appendix B for further details.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core contribution of the method does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.